Allen-Bradley

SLC 500™ BASIC and BASIC-T Modules

(Catalog Numbers 1746-BAS and 1746-BAS-T)

User Manual

Rockwell Automation

Important User Information

Because of the variety of uses for the products described in this publication, those responsible for the application and use of this control equipment must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards.

The illustrations, charts, sample programs and layout examples shown in this guide are intended solely for purposes of example. Since there are many variables and requirements associated with any particular installation, Rockwell International Corporation does not assume responsibility or liability (to include intellectual property liability) for actual use based upon the examples shown in this publication.

Rockwell Automation publication SGI-1.1, *Safety Guidelines for the Application, Installation and Maintenance of Solid-State Control* (available from your local Rockwell Automation office), describes some important differences between solid-state equipment and electromechanical devices that should be taken into consideration when applying products such as those described in this publication.

Throughout this manual we use notes to make you aware of safety considerations:

ATTENTION

Identifies information about practices or circumstances that can lead to personal injury or death, property damage or economic loss

Attention statements help you to:

- identify a hazard
- avoid a hazard
- recognize the consequences

IMPORTANT Identifies information that is critical for successful application and understanding of the product.

The information below summarizes the changes to this manual since the last printing.

To help you find new information and updated information in this release of the manual, we have included change bars as shown to the right of this paragraph.

New Information

References to the 1746-BAS-T module were added throughout the manual. Specific information regarding 1746-BAS-T optional memory modules and module installation is shown in the table below.

For this new information	See page(s)
1746-BAS-T Overview	1-1
Optional Memory Modules	2-2
Jumper Settings for Optional Memory Modules	3-3 and 3-4
1746-BAS-T Module ID Code	4-3

Appendix C

Lithium Battery Replacement, Handling, and Disposal

Read this preface to familiarize yourself with the rest of the manual. This preface covers the following topics:

- who should use this manual
- the purpose of this manual
- how to use this manual
- terms and abbreviations
- conventions used in this manual
- Rockwell Automation support

Who Should Use This Manual

Use this manual if you are responsible for designing, installing, programming, or troubleshooting control systems that use Allen-Bradley small logic controllers.

You should have a basic understanding of SLC 500™ products. You should understand programmable controllers and be able to interpret the ladder logic instructions required to control your application. If you do not, contact your local Rockwell Automation representative for information on available training courses before using this product.

Purpose of this Manual

This manual is a reference guide for the design and installation of the SLC 500 BASIC and BASIC-T modules. It describes the procedures for installing and using the modules.

Chapter	Title	Contents
	Preface	Describes the purpose, background, and scope of this manual. Also lists related publications.
1	Module and Development Software Overview	Explains the hardware and software features.
2	Component Selection	Explains and illustrates how to select memory modules, network configurations, and modems for your application.
3	Installing and Wiring your Module	Provides installation procedures and wiring guidelines.
4	Programming Overview	Provides information needed to program your module.
Appendix A	Specifications	Presents the modules' specifications.
Appendix B	Worksheets	Describes how to set the module for proper functioning.
Appendix C	Lithium Battery Replacement, Handling, and Disposal	Provides important information for the replacement, handling, and disposal of lithium batteries.

Related Documentation

The following documents contain additional information regarding Rockwell Automation products. To obtain a copy, contact your local Rockwell Automation office or distributor.

For	Read this document	Publication Number
A BASIC Language reference manual that describes BASIC commands, CALLS, and functions	BASIC Language Reference Manual	1746-RM001A-US-P
A programming manual with detailed instructions on installing and using BASIC Development Software to program the BASIC and BASIC-T modules.	BASIC Development Software Programming Manual	1746-PM001A-US-P
An overview of the SLC 500 family of products	SLC 500 System Overview	1747-SO001A-US-P
A description of how to install and use a Modular SLC 500 Processor	Modular Hardware Style Installation and Operation Manual	1747-6.2
A reference manual that contains status file data and instruction set information for SLC 500 processors	SLC 500™ and MicroLogix™ 1000 Instruction Set Reference Manual	1747-6.15
A description of how to install and use a module that acts as a bridge between DH485 networks and devices requiring DF1 protocol.	DH485/RS-232C Interface Module User's Manual	1747-6.12
In-depth information on grounding and wiring Allen-Bradley programmable controllers	Allen-Bradley Programmable Controller Grounding and Wiring Guidelines	1770-4.1
A glossary of industrial automation terms and abbreviations	Allen-Bradley Industrial Automation Glossary	AG-7.1
An article on wire sizes and types for grounding electrical equipment	National Electric Code	Published by the National Fire Protection Association of Boston, MA

How to Use this Manual

To use this manual effectively, use the worksheets provided in appendix B. The worksheets can help you document your application and settings and also facilitate the flow of information to other individuals in your organization for implementation.

Terms and Abbreviations

The following terms and abbreviations are specific to this product. For a complete listing of Allen-Bradley terminology, refer to the *Allen-Bradley Industrial Automation Glossary*, publication number ICCG-7.1.

- Module — SLC 500 BASIC and BASIC-T Modules (catalog numbers 1746-BAS and 1746-BAS-T)
- BASIC development software — BASIC Development Software (catalog number 1747-PBASE)
- DH485 — network communication protocol
- EPROM — Erasable Programmable Read Only Memory
- MTOP — system control value that holds the last valid memory address
- RS-232/423 — serial communication interface
- RS-422 — differential communication interface
- RS-485 — network communication interface
- SLC 500 — SLC 500 fixed and modular controller

Conventions Used in this Manual

The following conventions are used throughout this manual:

- Bulleted lists such as this one provide information, not procedural steps.
- Numbered lists provide sequential steps or hierarchical information.
- *Italic* type is used for emphasis.
- Text in **`this font`** indicates words or phrases you should type.
- Key names match the names shown and appear in bold, capital letters within brackets (for example, **`[ENTER]`**).

Rockwell Automation Support

Rockwell Automation offers support services worldwide, with over 75 Sales/ Support Offices, 512 authorized Distributors and 260 authorized Systems Integrators located throughout the United States alone, plus Rockwell Automation representatives in every major country in the world.

Local Product Support

Contact your local Rockwell Automation representative for:

- sales and order support
- product technical training
- warranty support
- support service agreements

Technical Product Assistance

If you need to contact Rockwell Automation for technical assistance, please review the information in the appropriate chapter first. Then call your local Rockwell Automation representative.

Your Questions or Comments on this Manual

If you find a problem with this manual or if you have any suggestions for how this manual could be made more useful to you, please contact us at the address below:

Rockwell Automation
Control and Information Group
Technical Communication, Dept. A602V
P.O. Box 2086
Milwaukee, WI 53201-2086

Module and Development Software Overview

This chapter introduces you to the SLC 500™ BASIC and BASIC-T modules and the BASIC Development Software. After reading this chapter you should be familiar with the:

- module components and features
- BASIC Development Software features
- typical configurations of the module
- module hardware specifications
- module-related products

Overview

The module and the development software provide the following benefits:

- easy data collection from user devices
- integrated program debugging environment
- operator interface capabilities
- flexible program and data storage options
- high-level math
- clock/calendar
- high-level programming environment
- extensive online help system
- easy access to editor functions through user interface
- advanced text editor windows

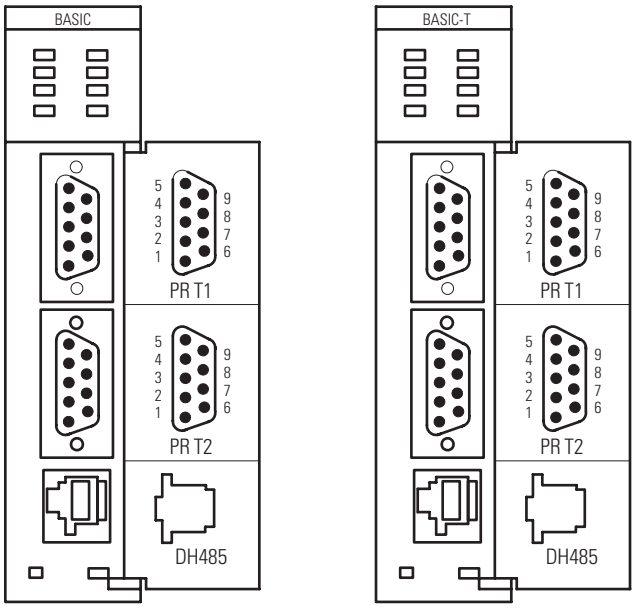
NOTE The 1746-BAS-T is a higher-speed version of the 1746-BAS module with identical hardware features. The modules can be interchanged, except that the 1746-BAS-T uses different (optional) memory modules. Due to the high speed of the 1746-BAS-T, existing programs written for the 1746-BAS may require adjustment for identical operation using the faster 1746-BAS-T module.

BASIC and BASIC-T Modules

The modules are single-slot modules that reside in a SLC 500 fixed or modular controller chassis. Use the module as:

- a foreign device interface
- an operator interface

Figure 1.1 Module with Door Open



Hardware Features

The module provides the following hardware features:

- 24K bytes of battery backed RAM for storage of user programs and data
- capacitive backup of RAM during battery change
- socket for a standard 1747-M1, M2, M3, or M4 memory module (1746-BAS) for non-volatile storage of user programs
- socket for a 1771-DBMEM1 or -DBMEM2 memory module (1746-BAS-T) for non-volatile storage of user programs
- battery-backed, 24-hour clock/calendar
- free-running clock with 5 ms resolution
- two isolated 9-pin D-shell serial ports (PRT1 and PRT2) that provide RS-232/423, RS-422, and RS-485 communication with I/O devices
- Port PRT2 provides DF1 full-duplex or half-duplex slave protocol for SCADA applications
- one RJ-45 port (DH485) that provides communication over the DH485 network
- multiple LEDs for operator interface
- SLC 500 backplane interface

Software Features

The module provides the following software features:

- BASIC programming with the Intel BASIC-52 Language and enhancements
- SLC 500 backplane data read and write support including image table transfers and M0 and M1 file transfers
- execution of programs from memory modules
- string manipulation support
- DH485 network support
- DF1 protocol support
- full set of trigonometric function instructions
- floating point calculations and conversion
- extensive call libraries

Module Communication Ports

There are three communication ports on the front of the module. The location, name, and pin numbers of these ports are listed on the inside of the module door. They are:

- PRT1 – Used to interface the module with user devices. This port is a serial port that accommodates RS-232/423, RS-422, and RS-485 communication modes. Port PRT1 is capable of operating full-duplex at 300, 600, 1200, 2400, 4800, 9600, and 19200 baud. The default settings are 1200 baud, RS-232/423 communications.
- PRT2 – Used to interface the module with user devices or a modem using DF1 protocol. This port is a serial port that accommodates RS-232/423, RS-422, and RS-485 communication modes. Port PRT2 is capable of operating full-duplex at 300, 600, 1200, 2400, 4800, 9600, and 19200 baud.
- DH485 – Used to interface the module with the DH485 network. This port is not isolated and cannot directly drive the DH485 network. You must use a 1747-AIC link coupler to link port DH485 with the DH485 network.

IMPORTANT When DF1 protocol is selected on port PRT2, DH485 communications are disabled.

Module LEDs

There are eight LEDs on the front of the module. These LEDs are used for module diagnostics and operator interface. The LEDs and their indications are shown on page 1-4.

Figure 1.2 Module LEDs

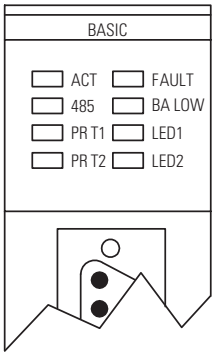


Table 1.1 Module LEDs

LED	Color	Status	Indication
ACT	Green	ON	The module is receiving power from the backplane and is executing BASIC code.
		Blinking	The module is in Command mode.
		OFF	The module is not receiving power from the backplane. A fault condition exists.
485	Green	ON	Port DH485 on the module is active for communication.
		OFF	Port DH485 on the module is not active for communication.
PRT1	Green	Blinking	Port PRT1 on the module is transmitting or receiving signals.
		OFF	Port PRT1 on the module is not transmitting or receiving signals.
PRT2	Green	Blinking	Port PRT2 on the module is transmitting or receiving signals.
		OFF	Port PRT2 on the module is not transmitting or receiving signals.
FAULT	Red	ON	A system problem was detected during background diagnostics. Contact your local Allen-Bradley representative.
		OFF	No system problems are detected during background diagnostics.
BA LOW	Red	ON	The voltage of the battery that backs up RAM is low. A new battery is needed.
		OFF	The voltage of the battery that backs up RAM is at an acceptable level.
LED1	Amber	ON	User definable. LED activated through the user program.
		OFF	User definable. LED de-activated through the user program.
LED2	Amber	ON	User definable. LED activated through the user program.
		OFF	User definable. LED de-activated through the user program.

BASIC Development Software (1747-PBASE)

The BASIC Development Software provides the user with a structured and efficient means to create BASIC programs for the module. This software is loaded into a an MS-DOS compatible personal computer. It uses the personal computer to facilitate editing, compiling (translating), uploading, and downloading of BASIC programs.

The BASIC Development Software has a menu-driven, window-type environment that offers:

- pull-down menus to access all editor functions
- function key access to frequently used functions
- multiple window editing
- cut and paste support between windows
- search and replace support
- search between files support
- built-in calculator that can paste results into your program
- ASCII look-up table
- line draw editor to create operator interface images without having to enter ASCII characters
- keystroke macros
- undo and redo functions
- extensive help messages for each menu, menu option, and for keywords embedded in the menu text
- capability to create user-defined macro libraries
- sophisticated debug tools including watch windows, single-step operation, and go to cursor breakpoint operation
- syntax checked translations to native BASIC to reduce debug time
- BASIC translator that steps through the BASIC program and identifies errors
- ASCII terminal mode
- hex file transfer support

The development software enables you to program the module from a personal computer connected to either the module's DH485 or PRT1 ports. The software allows direct access to the module through terminal emulation over an RS-232/423 or DH485 network.

Refer to the *BASIC Development Software Programming Manual* (publication number 1746-PM001A-US-P) for additional information on the software.

Typical Configurations

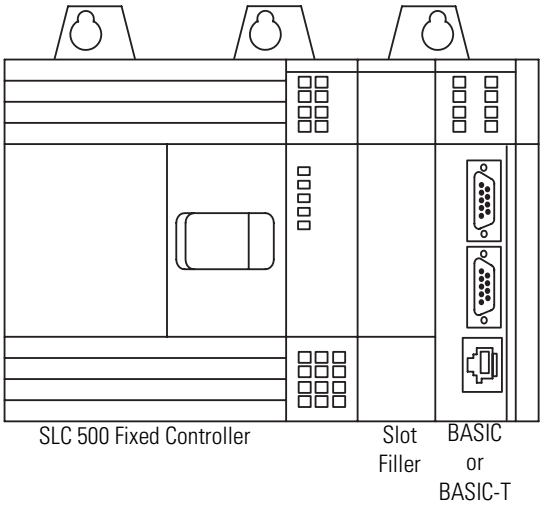
The typical configuration of the SLC system that incorporates your BASIC or BASIC-T module depends on whether the module is:

- integrated with a SLC 500 fixed or modular controller
- programmed directly with an ASCII terminal or programmed using a personal computer with the BASIC Development Software 1747-PBASE
- communicating with a DH485 network or with an external source through a modem using DF1 protocol

Module Integration

The module is a single-slot module that is inserted into a slot in the expansion chassis of your SLC 500 fixed controller or an open slot in the 1746 I/O chassis of your SLC 500 modular controller. The module may be inserted in any slot of the 1746 I/O chassis except the first slot of the first chassis, which is reserved for the SLC modular processor. Typical SLC fixed and modular configurations are shown in the following figures.

Figure 1.3 Typical Configurations



Module Programming Interface

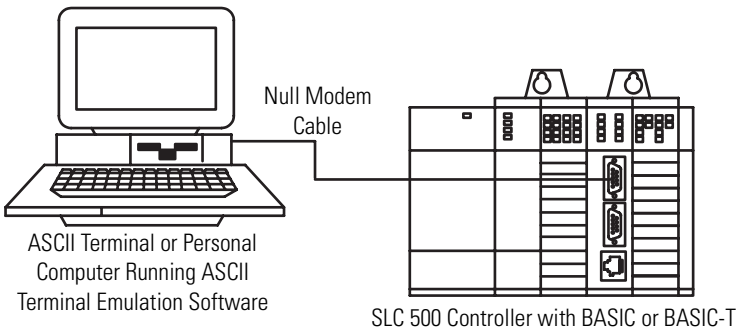
Your module can be programmed using an ASCII terminal with ASCII terminal emulation software. You can also use a personal computer with the BASIC Development Software (catalog number 1747-PBASE). Use an ASCII terminal to enter a BASIC program one line at a time to the module. Use a personal computer with the BASIC Development Software to create a BASIC program that is then downloaded to your module. Typical programming interface configurations include:

- ASCII terminal interface – Figure 1.4
- BASIC Development Software interface (RS-232) – Figure 1.5
- BASIC Development Software interface (DH485) – Figure 1.6

ASCII Terminal Interface

Use an ASCII terminal to enter a BASIC program one line at a time to your module through port PRT1. The ASCII terminal connected to the module must be an industrial terminal, workstation, or personal computer (without the BASIC Development Software) that communicates in alphanumeric mode. An ASCII terminal can also be used to display charts or graphs generated by the BASIC program. Figure 1.4 shows a typical ASCII terminal interface.

Figure 1.4 Module ASCII Terminal Interface

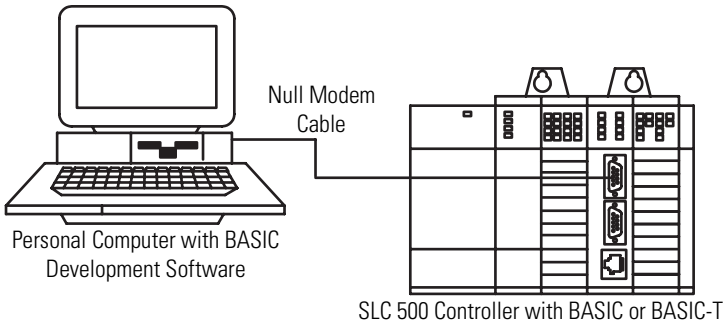


In this configuration, the RS232 port on the back of your industrial terminal or personal computer is connected to port PRT1 on your module. Port PRT1 must be configured as the program port. See chapter 3 for additional information on port configuration.

BASIC Development Software Interface (RS232)

Use a personal computer with the BASIC Development Software (PBASE) to create a BASIC program that is then downloaded to your module. PBASE provides an efficient means to edit, compile (translate), upload, and download BASIC programs. Refer to the *BASIC Development Software Programming Manual* (publication number 1746-PM001A-US-P) for additional information on this software.

Figure 1.5 BASIC Development Software Interface (RS-232)



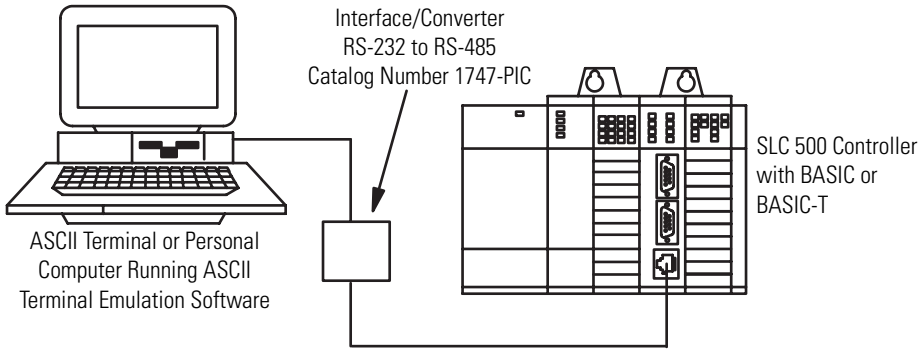
In this configuration, the serial port on the personal computer is connected to port PRT1 on the module. The personal computer communicates with the module through terminal emulation over an RS-232 interface. Port PRT1 must be configured as the program port. See chapter 3 for additional information on port configuration.

IMPORTANT When using the BASIC Development Software to interface with the RS-232 port of the module, PBASE must be configured for RS-232 communication through the configuration and terminal selection menus. Refer to the *BASIC Development Software Programming Manual* (publication number 1746-PM001A-US-P) for additional information.

BASIC Development Software Interface (DH485)

In this configuration, the serial port on the personal computer interfaced with port DH485 on the module through a 1747-PIC Interface/Converter. Port DH485 must be configured as the program port with DH485 protocol. See chapter 3 for additional information on port configuration.

Figure 1.6 BASIC Development software Interface (DH485)



The 1747-PIC Interface/Converter converts the RS-232 signals from the personal computer RS-232 serial port to RS-485 format.

IMPORTANT When using the BASIC Development Software to interface with port DH485 of the module, PBASE must be configured for DH485 communication through the configuration and terminal selection menus. Refer to the *BASIC Development Software Programming Manual* (publication number 1746-PM001A-US-P) for additional information.

Module Network Configurations

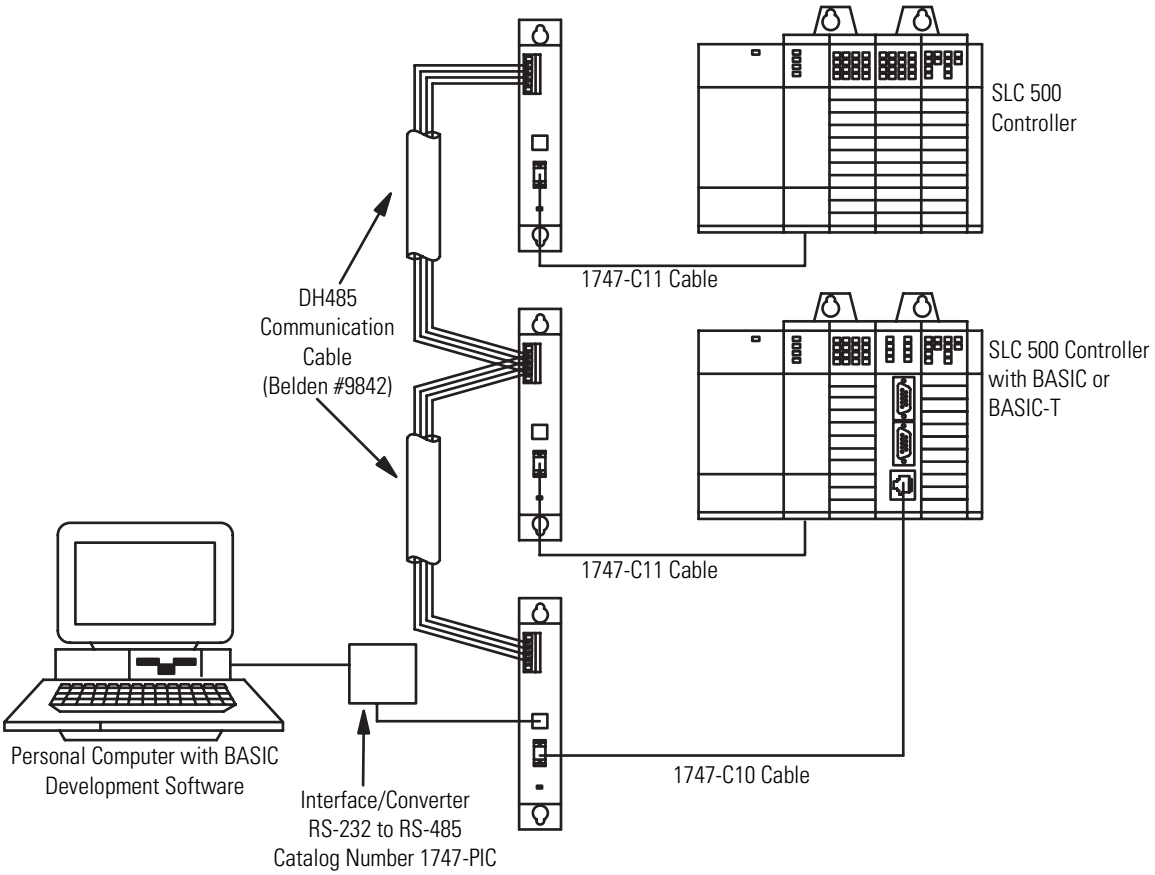
Your module may communicate with a DH485 network or it can communicate with a remote device through a modem using the DF1 protocol. When DF1 protocol is used on PRT2, port DH485 is disabled. Typical communication configurations are shown in the following figures:

- DH485 network configurations – Figure 1.7, Figure 1.8, and Figure 1.10
- DF1 protocol configuration – Figure 1.10

ATTENTION Do not place the module on an active DH485 network until the node address and baud rate of the module are configured. Refer to the *BASIC Language Reference Manual* (publication number 1746-RM001A-US-P) for additional information on setting the module node address and baud rate.

Figure 1.7 shows the module interfaced with a DH485 network through a 1747-AIC Isolated Link Coupler. The link coupler also provides an interface to the DH485 network for a personal computer with the BASIC Development Software.

Figure 1.7 DH485 Network Configuration



The 1747-PIC Interface/Converter converts the RS-232 signals from the personal computer RS-232 serial port to RS-485 format. The 1747-AIC link coupler links the converted signals with the DH485 network and port DH485 on the module. Port DH485 must be configured as the program port in order to communicate with PBASE software via the DH485 network. See chapter 3 for additional information on port configuration.

IMPORTANT

- Each module requires a link coupler port to interface it with the DH485 network.
- When using the BASIC Development Software to interface with the module, the BASIC Development Software must be configured for DH485 communication through the configuration and terminal selection menus. Refer to the *BASIC Development Software Programming Manual* (publication number 1746-PM001A-US-P) for additional information.

Figure 1.8 also shows the module interfaced with a DH485 network through a 1747-AIC link coupler. The link coupler also provides an interface to the DH485 network for a personal computer with the BASIC Development Software. In this configuration, a 1784-KR DH485 Interface Card must be installed in the personal computer.

IMPORTANT Type **PBASE/KR** to select the driver software. This allows you to interface directly to the DH485 network when using a 1784-KR DH485 Interface Card in your personal computer.

Figure 1.8 DH485 Network Configuration Using a 1747-C10 Cable

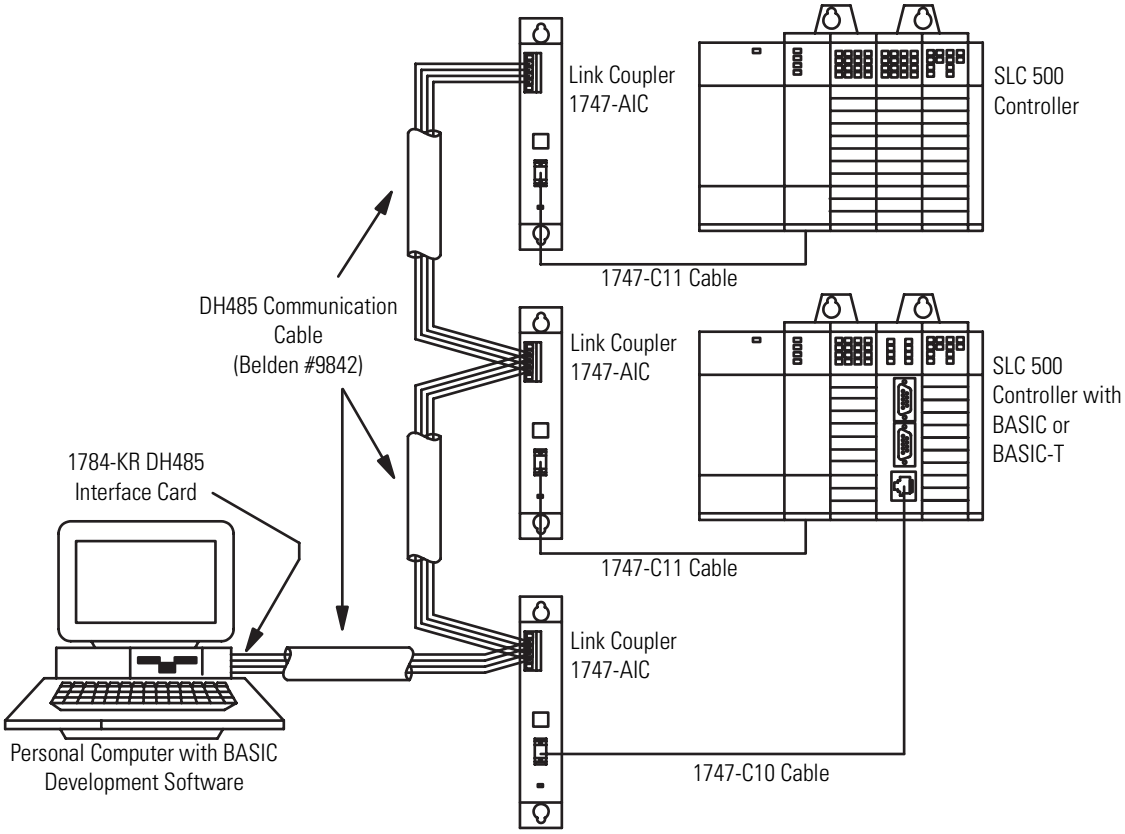
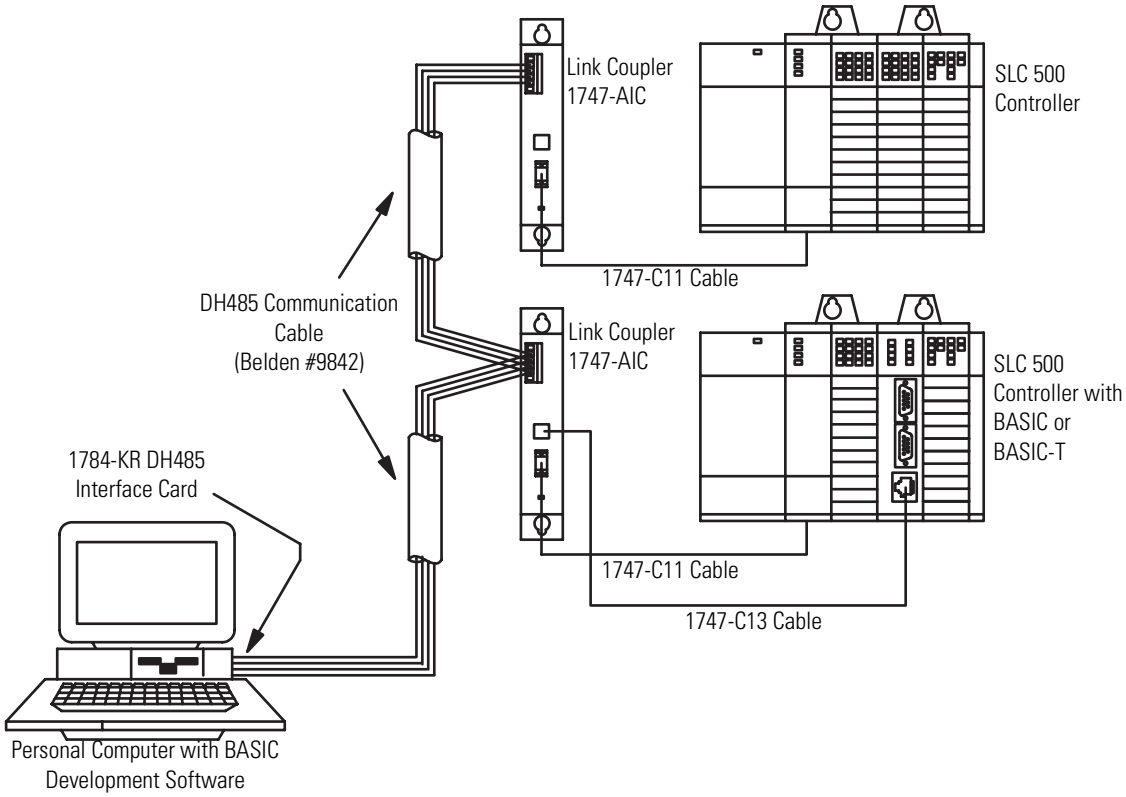


Figure 1.9 DH485 Network Configuration Using a 1747-C13 Cable



IMPORTANT When the 1747-C13 cable is used, you can connect the center port on any link coupler on the network directly to the module. This may reduce the number of link couplers that you need. (Compare the amount of equipment shown in figures 1.8 and 1.9.)

The 1747-C13 cable acts only as a communication link and does not carry 24V dc power. The 1747-C10 or 1747-C11 cable carries 24V dc power from the processor to the link coupler as shown above. (The 1747-C10 cable and 1747-C11 cable are interchangeable.)

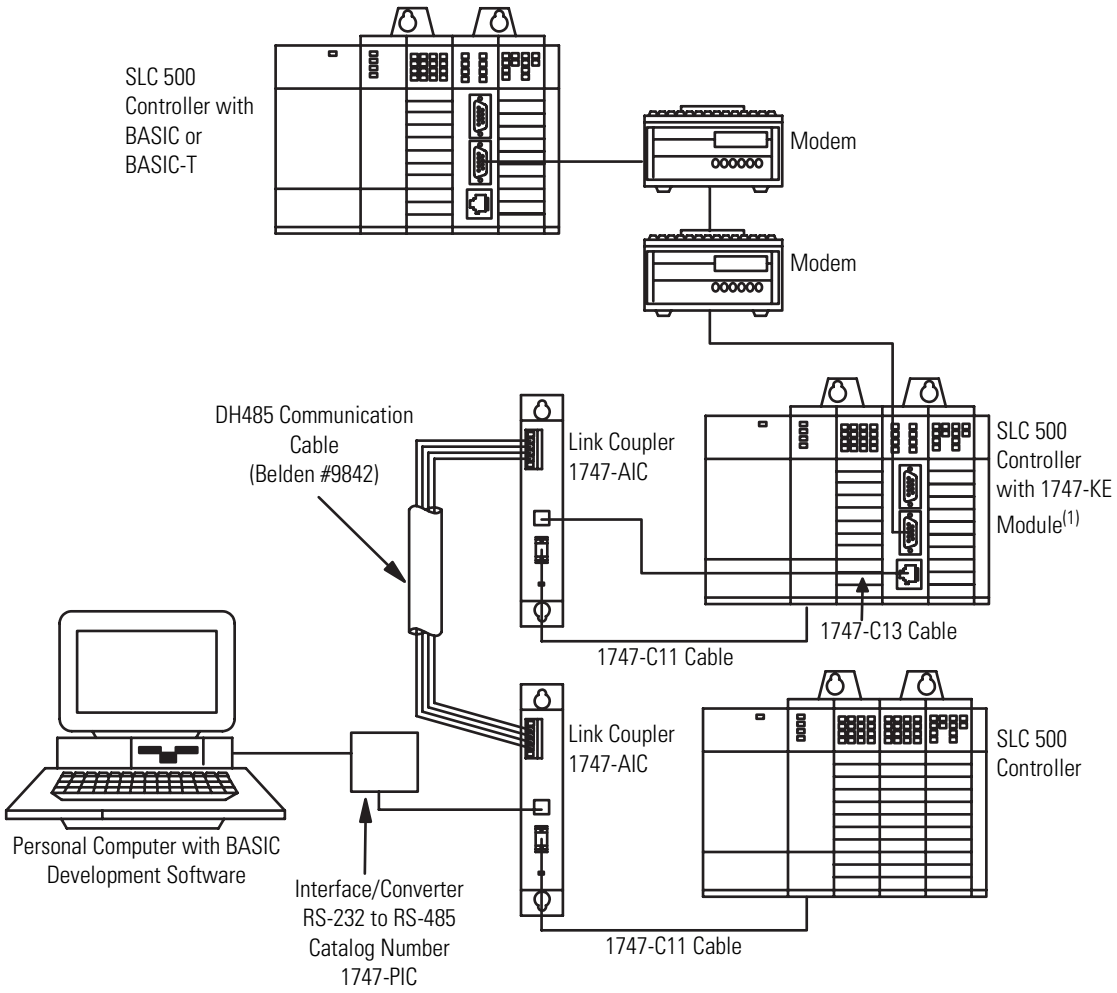
You can also supply power to the link coupler by connecting the 1747-C10 or 1747-C11 cable from the link coupler to the module and not to the processor.

The 1784-KR DH485 Interface Card enables the personal computer to communicate with the DH485 network. The DH485 data link connector on the 1784-KR card and port DH485 on your module are interfaced with the DH485 network through a 1747-AIC link coupler.

Port DH485 must be configured as the program port in this configuration. See chapter 3 for additional information on port configuration.

Figure 1.10 shows the module using DF1 to control communications with a modem. In this configuration, the module is interfaced with a DH485 network through a peer-to-peer communication interface with full-duplex, DF1 protocol.

Figure 1.10 Full-Duplex, DF1 Protocol Configuration



(1) The 1747-KE module can be replaced by the 1770-KF3 DH485 Communication Interface Module. The KF3 is a stand-alone version of the KE module.

The modems in this configuration may be dial-up. If the modems are dial-up, the BASIC program may initiate dial-up and then switch port PRT2 to DF1 protocol when connection is made to the 1747-KE or 1770-KF3 DH485 Communication Interface Module. Port PRT2 on your module must be configured as having DF1 protocol. See chapter 3 for additional information on port configuration.

The 1747-C13 Cable acts only as a communication link and does not carry 24V dc power. The 1747-C10 or 1747-C11 Cable carries 24V dc power from the processor to the link coupler. (The 1747-C10 Cable and 1747-C11 Cable are interchangeable.)

IMPORTANT By configuring JW4 for DF1 communication on PRT2, DH485 communications are disabled.

Component Selection

After reading this chapter, you should understand the:

- module memory requirements for BASIC programming and be able to select the memory modules necessary for your application
- concepts of connecting your module to the DH485 network and be able to select the components necessary for your application
- concepts of connecting a modem to the module and be able to select the necessary components for your application

Memory Requirements for BASIC Programming

The module uses the following types of memory modules during BASIC programming:

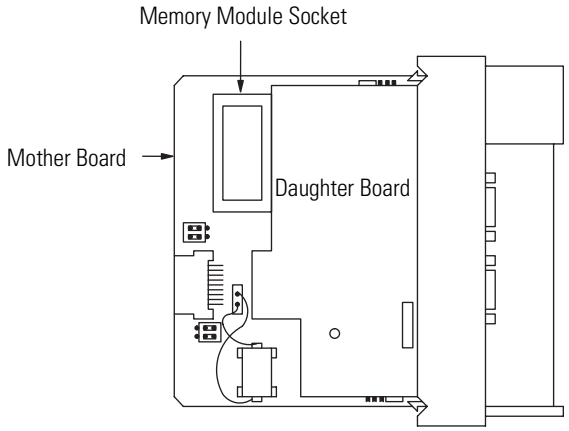
- a 32K byte battery backed RAM of which 24K byte is reserved to store BASIC programs and protected variables
- an optional 8K or 32K byte memory module to store BASIC programs

IMPORTANT Maximum user program storage space is 56K byte. Battery backed RAM provides 24K byte and the optional memory module provides up to 32K byte.

Optional Memory Module

The optional memory module provides non-volatile storage of user BASIC programs and port configuration. The socket that holds the optional memory module is located on the module's mother board as shown in Figure 2.1.

Figure 2.1 Optional Memory Module Socket Location



ATTENTION

Do not expose the module to surfaces or other areas that may typically hold an electrostatic charge. Electrostatic charges can alter or destroy memory.

You may use any of the following memory module options with your module:

- 1747-M1, 8K byte EEPROM (1746-BAS only)
- 1747-M2, 32K byte EEPROM (1746-BAS only)
- 1747-M3, 8K byte UVPROM (1746-BAS only)
- 1747-M4, 32K byte UVPROM (1746-BAS only)
- 1771-DBMEM1, 8K byte EEPROM (1746-BAS-T only)
- 1771-DBMEM2, 32K byte EEPROM (1746-BAS-T only)

Your module can program the 1747-M1, 1747-M2, 1771-DBMEM1, and DBMEM2 EEPROM optional memory modules. The 1747-M3 and 1747-M4 UVPROM optional memory modules must be programmed by an external PROM programmer. Jumper JW3 is used to redirect the module circuitry for the different memory module options. Refer to chapter 3 of this manual for additional information on jumper JW3.

IMPORTANT The module can program and erase EEPROM memory modules. However, it cannot program or erase UVPROM memory modules. Refer to the *SLC 500 Fixed and Modular Style Programmable Controller's Installation and Operation Manuals* (catalog numbers 1747-6.1 and 1747-6.2) for additional information on programming and erasing UVPROMs.

The data format of the module EEPROM and UVPROM optional memory modules is hexadecimal. The BASIC development software provides a hex file transfer option that can be used to upload and download hex files to the module EEPROM or UVPROM. The primary use of hex file transfers is to transfer the data from an EEPROM in one module to an EEPROM in another module. Hex file transfers can also be used to copy the data of an EEPROM to a UVPROM via a PROM programmer. Refer to the *BASIC Development Software Programming Manual* (publication number 1747-PM001A-US-P) for additional information on hex file transfers.

Components Required for DH485 Communication

Your module and the BASIC development software can interface with a DH485 network using a combination of the following:

- 1747-AIC Isolated Link Coupler
- 1747-PIC Interface/Converter
- 1784-KR DH485 Interface Card
- 1770-KF3 DH485 Communication Interface Module, or
- 1747-KE DH485/RS-232C Communication Interface Module

Figure 2.2 and Figure 2.3 show some of these components in typical DH485 network interfaces.

Figure 2.2 DH485 Network Interface

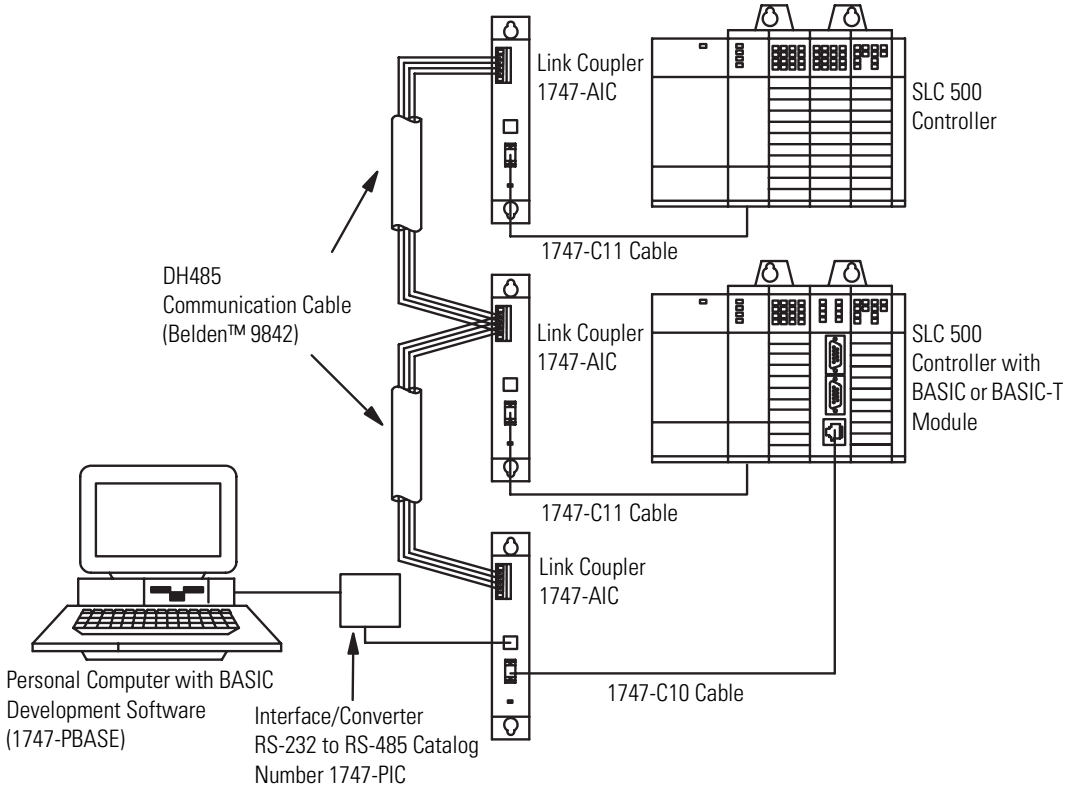
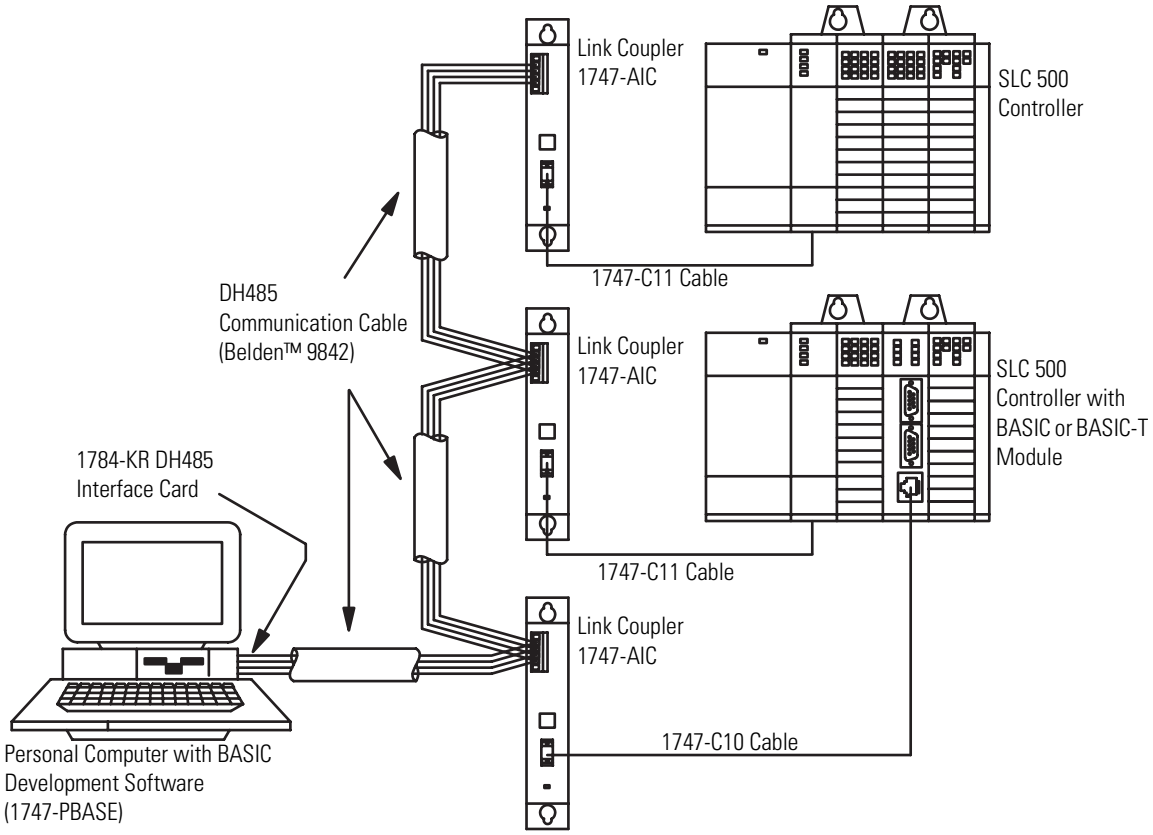


Figure 2.3 DH485 Network Interface



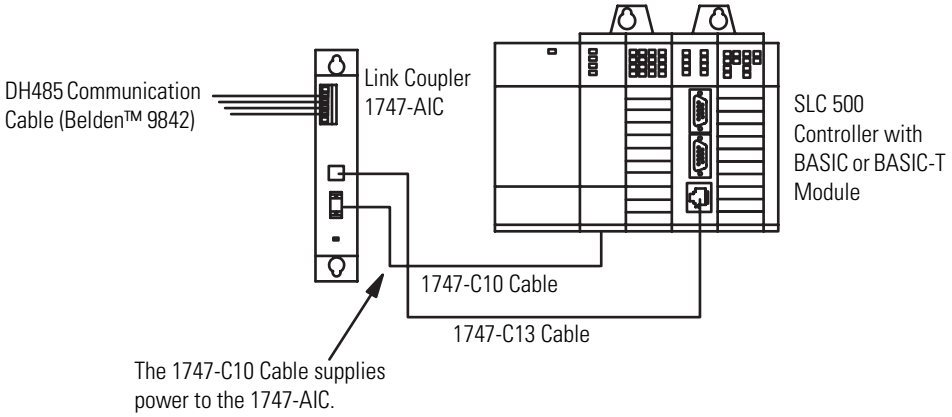
Alternate Connection

The 1747-C13 Cable is designed to connect the following SLC 500 products to each other:

- 1747-AIC Isolated Link Coupler
- a 1746-BAS or 1746-BAS-T BASIC Module
- fixed controllers
- modular controllers

The following figure shows one application for the 1747-C13 Cable.

Figure 2.4 1747-C13 Cable Connection



The 1747-C13 cable acts only as a communication link and does not carry 24V dc power. The 24V dc can come from either the processor or an outside power source. The 1747-C10 or 1747-C11 cable carries 24V dc power from the processor to the link coupler. (The 1747-C10 and 1747-C11 are interchangeable.) A cable connected to the outside power source carries 24V dc from the outside power source to the link coupler.

The module and SLC 500 CPU act as two separate nodes on the DH485 network.

Refer to the following sections for additional information on some of the components shown in the previous figures.

1747-AIC Isolated Link Coupler

The 1747-AIC Isolated Link Coupler allows you to link modules to the DH485 network. Figure 2.2 and Figure 2.3 show a DH485 network with the module and a personal computer linked to the network through a 1747-AIC link coupler.

1747-PIC Interface/Converter

Use the 1747-PIC Interface/Converter to convert the RS-232 signals from the personal computer's serial port to RS-485 signal format. Figure 2.2 shows the interface/converter integrating a personal computer with the BASIC development software to the module across a DH485 network.

IMPORTANT When using the BASIC development software to interface with the module through the 1747-PIC, the BASIC development software must be configured for DH485 communication through the configuration and terminal selection menus. Refer to the *BASIC Development Software Programming Manual* (publication number 1746-PM001A-US-P) for additional information.

1784-KR DH485 Interface Card

The 1784-KR DH485 Interface Card enables your personal computer to communicate across the DH485 network to the module without the interface/converter. Figure 2.3 shows a DH485 network configuration with the 1784-KR DH485 Interface Card and its host computer linked with the module through a link coupler.

In this configuration, your personal computer must have the 1784-KR DH485 Interface Card installed in one of its expansion slots. The DH485 data link connector on the 1784-KR card and port DH485 on your module are interfaced with the DH485 network through a 1747-AIC link coupler. Port DH485 must be configured as the program port in this configuration.

DH485 Cable Requirements

Use the 1747-C10 cable, 1747-C11 cable, or 1747-C13 cable to interface port DH485 of the module with a 1747-AIC link coupler. Use the DH485 communication cable, Belden #9842, to interface between the link couplers on the DH485 network.

Refer to the *SLC 500 Fixed and Modular Style Programmable Controller's Installation and Operation Manuals* (publication numbers 1747-6.1 and 1747-6.2) for additional information on DH485 cables.

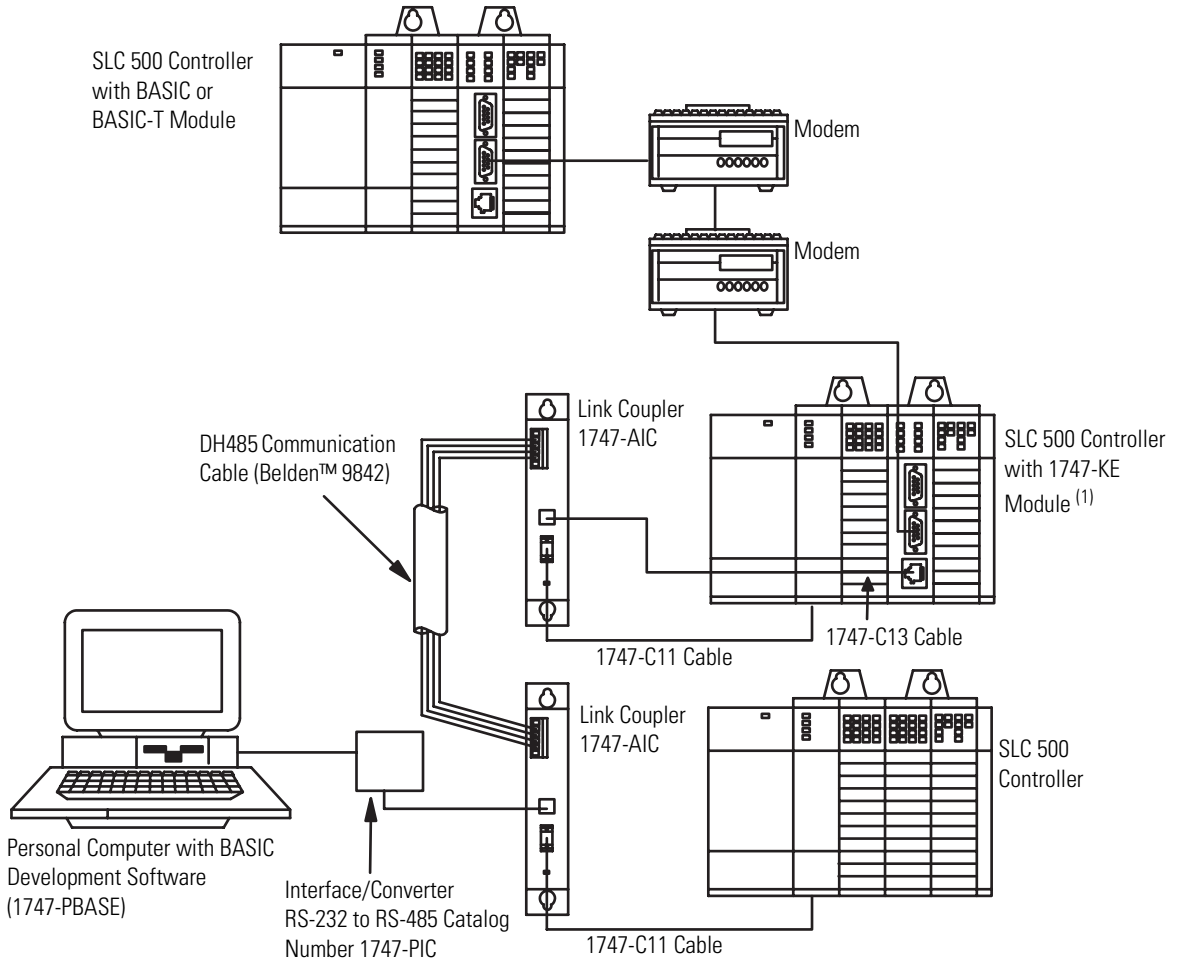
Components Required for DF1 Communication

Your module may use DF1 to communicate with external devices. The DF1 driver is accessed through port PRT2. The module communicates with the external devices using one of the following:

- a leased phone line
- a radio link
- a dial-up modem

Refer to the following sections for additional information on these components.

Figure 2.5 DF1 Communication Interface



(1) The 1747-KE module can be replaced by the 1770-KF3 DH485 Communication Interface Module, a stand-alone version of the KE module.

Leased Phone Line

A leased phone line is a private, dedicated phone line. Leased phone lines provide a phone link between modems that is available for communication at all times. Typically, leased phone lines are used when you have a high or constant transfer of communication between the module and external devices.

Radio Link

A radio link provides a communication link when phone lines are inaccessible or expensive to use. A radio link provides a communications link between radio modems.

Dial-Up Modem

Dial-up or phone modems are modems capable of communicating across standard phone lines. One dial-up modem initiates the communication, while another modem receives the communication.

Installing and Wiring Your Module

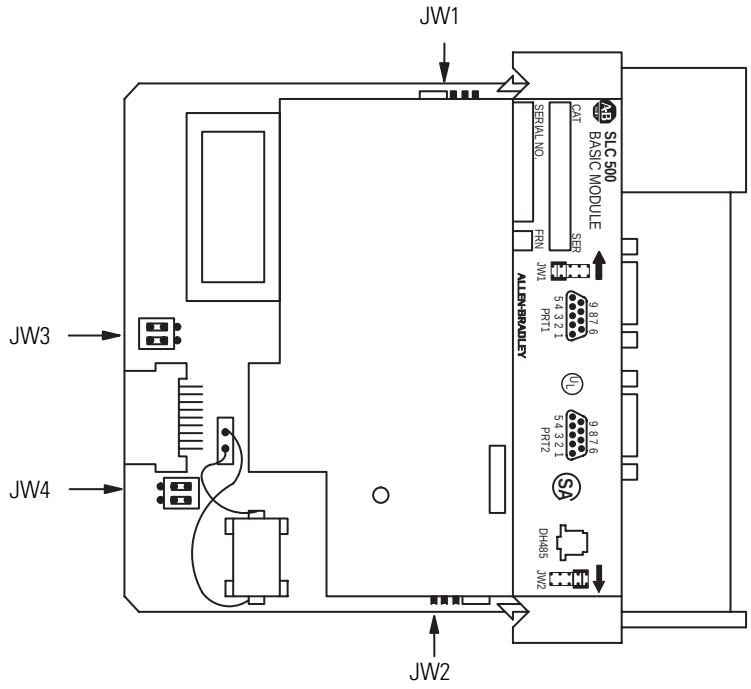
After reading this chapter, you should be able to:

- set the module's jumpers
- install your module into your SLC 500 fixed or modular controller system
- wire the mating connectors of the cables used to interface user devices to the module ports

Setting Module Jumpers

The module has four sets of jumpers that you need to set. Jumpers JW1 and JW2 configure ports PRT1 and PRT2. Jumper JW3 configures the type of optional memory module. Jumper JW4 configures the program port. Figure 3.1 shows the location of these jumpers.

Figure 3.1 Jumper Locations



ATTENTION

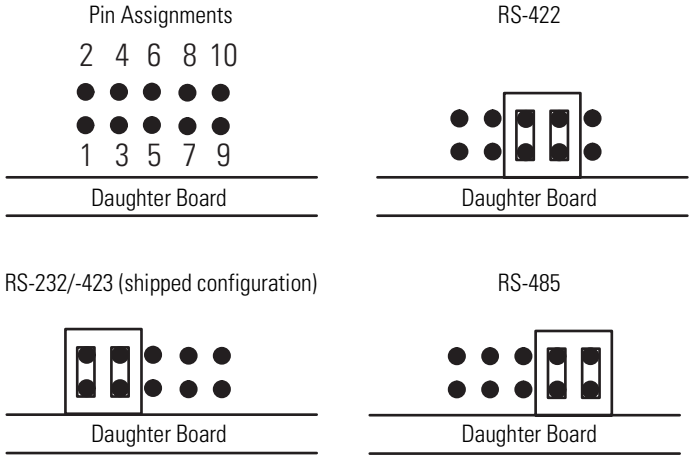
Do not expose the module to surfaces or other areas that may typically hold an electrostatic charge. Electrostatic charges can alter or destroy memory.

Setting Jumper JW1

Use jumper JW1 to select one of the following configurations for port PRT1:

- RS-232/423
- RS-422
- RS-485

Figure 3.2 JW1 Pin Assignments and Settings



ATTENTION

All other jumper settings for JW1 are illegal and may cause damage to the module.

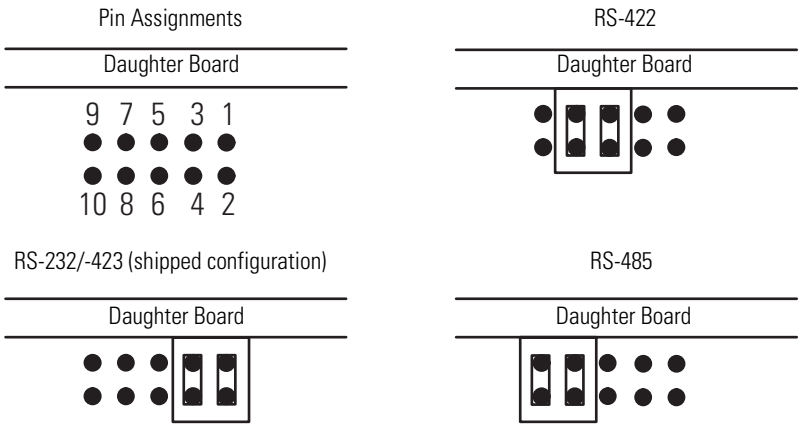
Use the worksheet in appendix B to document the selected jumper setting of jumper JW1. Documenting your selection provides others with information necessary to integrate the module with their SLC 500 fixed or modular controllers.

Setting Jumper JW2

Use jumper JW2 to select one of the following configurations for port PRT2:

- RS-232/423
- RS-422
- RS-485

Figure 3.3 JW2 Pin Assignments and Settings



ATTENTION

All other jumper settings for JW2 are illegal and may cause damage to the module

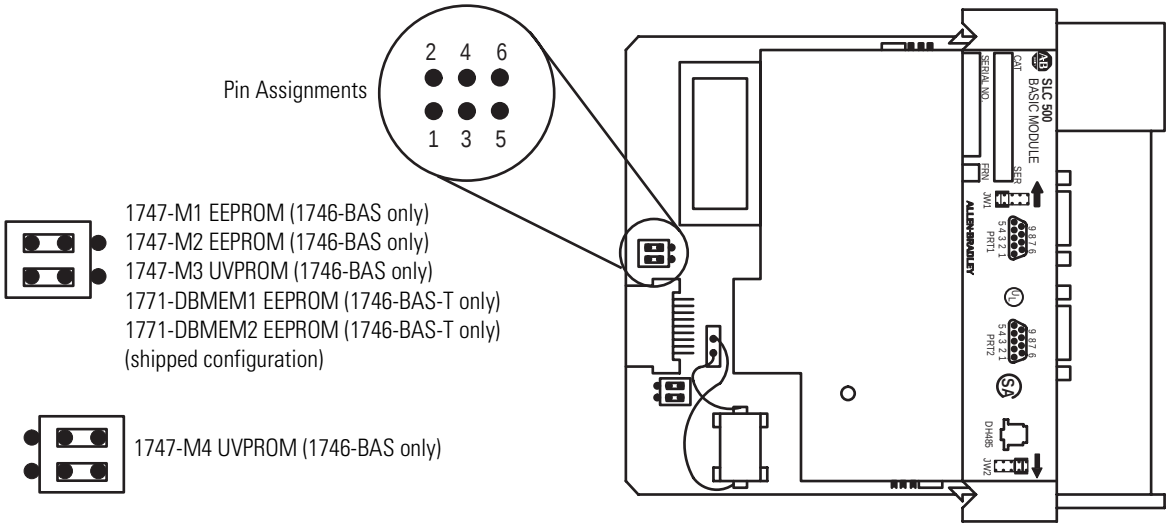
Use the worksheet in appendix B to document the selected jumper setting of jumper JW2. Documenting your selection provides others with information necessary to integrate the module with their SLC 500 fixed or modular controllers.

Setting Jumper JW3

Use jumper JW3 to configure the memory module socket for one of the following optional memory modules:

- 1747-M1, 8K bytes EEPROM (1746-BAS only)
- 1747-M2, 32K bytes EEPROM (1746-BAS only)
- 1747-M3, 8K bytes UVPROM (1746-BAS only)
- 1747-M4, 32K bytes UVPROM (1746-BAS only)
- 1771-DBMEM1, 8K bytes EEPROM (1746-BAS-T only)
- 1771-DBMEM2, 32K bytes EEPROM (1746-BAS-T only)

Figure 3.4 JW3 Pin Assignments and Settings



ATTENTION

All other jumper settings for JW3 are illegal and may cause damage to the module.

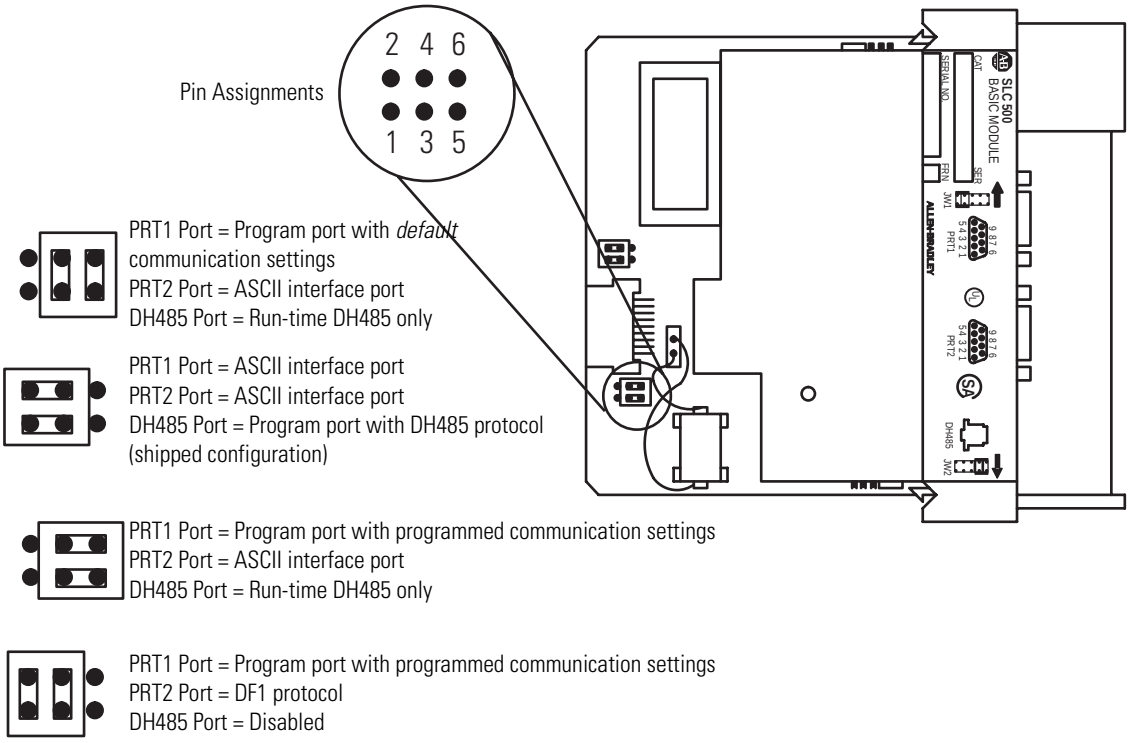
Use the worksheet in appendix B to document the selected jumper setting of jumper JW3. Documenting your selection provides others with information necessary to integrate the module with their SLC 500 fixed or modular controllers.

Setting Jumper JW4

Use jumper JW4 to select one of the following configurations for the module ports:

- PRT1 Port – Program port with *default* communication settings
  PRT2 Port – ASCII interface port
  DH485 Port – Run time DH485 operation only
- PRT1 Port – ASCII interface port
  PRT2 Port – ASCII interface port
  DH485 Port – Program port with DH485 protocol
- PRT1 Port – Program port with programmed communication settings
  PRT2 Port – ASCII interface port
  DH485 Port – Run time DH485 operation only
- PRT1 Port – Program port with programmed communication settings
  PRT2 Port – DF1 protocol
  DH485 Port – Disabled

Figure 3.5 JW4 Pin Assignments and Settings



IMPORTANT The first setting shown above is the default configuration. When the jumper is set in this position, the module always powers up in *Command mode* at 1200 baud, no parity, 8 data bits, and 1 stop bit.

ATTENTION

All other jumper settings for JW4 are illegal and may cause damage to the module.

IMPORTANT

- When DF1 protocol is selected for port PRT2, port DH485 is not available for DH485 programming or run-time operation.
- DF1 communication must be enabled through the BASIC program.

Use the worksheet in appendix B to document the selected jumper setting of jumper JW4. Documenting your selection provides others with information necessary to integrate the module with their SLC 500 fixed or modular controllers.

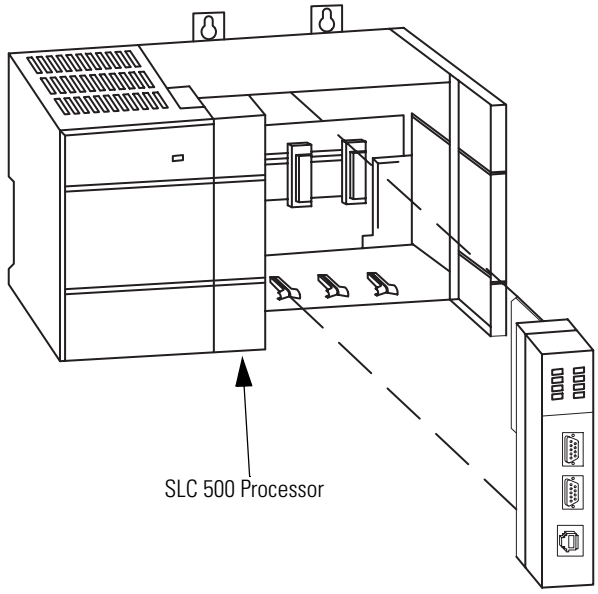
Installing Your module

Once you have unpacked and set the jumpers on your module, you are ready to install it in your:

- SLC 500 fixed controller expansion chassis
- SLC 500 modular controller 1746 I/O chassis

Your module may be installed in any open slot of your SLC 500 I/O chassis except the first slot of the first chassis, which is reserved for the processor module.

Figure 3.6 Installation in a SLC 500 I/O Chassis



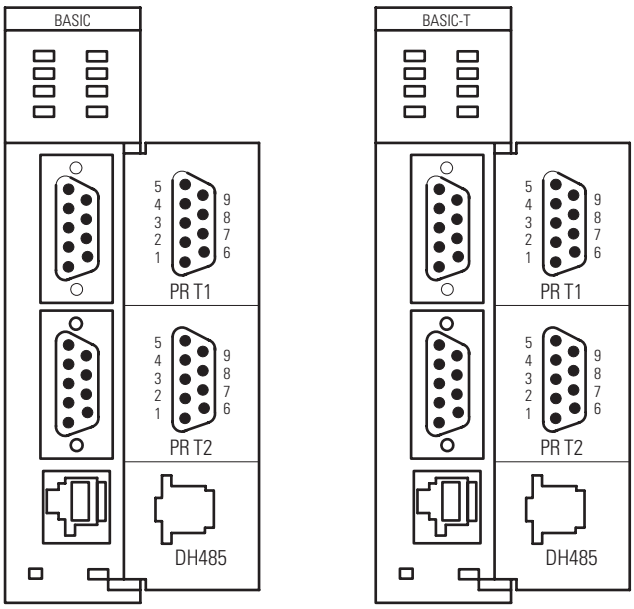
ATTENTION

Never install, remove, or wire any module with power applied to the chassis.

Wiring Your Communication Ports

The locations of the module's communication ports, PRT1, PRT2, and DH485, are shown in Figure 3.7.

Figure 3.7 Communication Ports



Wiring to Ports PRT1 and PRT2

Ports PRT1 and PRT2 can communicate to user devices through RS-232/423, RS-422, and RS-485 communication modes. Set jumpers JW1 and JW2 to reflect the communication mode you desire. The table below lists the pin assignments for ports PRT1 and PRT2.

Refer to the MODE command in the *BASIC Language Reference Manual*, publication number 1746-RM001A-US-P, for the default programming port configuration information

.

IMPORTANT When default communications are selected via JW4, the module defaults to the Command mode on powerup. Refer to page 3-4 of this manual for the default communication settings.

Use these pin assignments to wire the mating connector of the cable used to interface a user device to port PRT1. The sockets of this connector must be wired to correspond to the selected communication mode.

Table 3.1 Ports PRT1 and PRT2 Pin Assignments

Pin	RS-232/423	RS-422	RS-485	IBM AT Standard RS-232 Signals
1	Note 1	422 TXD -	TRXD -	DCD or CD
2	RXD	422 RXD -	(3)	RXD
3	TXD	(2)	(2)	TXD
4	DTR	(2)	(2)	DTR
5	COMMON	COMMON	COMMON	COMMON
6	DSR	422 RXD +	(3)	DSR
7	RTS	(2)	(2)	RTS
8	CTS	(2)	(2)	CTS
9	(1)	422 TXD +	TRXD +	RI

(1) In RS-423 mode, these pins are still connected to their RS-422 loads. Do not use these pins in RS-423 mode.

(2) In RS-422 and RS-485 modes these pins are connected to their RS-423 drivers and receivers. Do not use these pins in either RS-422 or RS-485 mode.

(3) In RS-485 mode, these pins are still connected to their RS-422 receivers. Do not use these pins in RS-485 mode.

Wiring diagrams for the RS-232/423 communication mode are shown starting on page 3-10.

Hardware Handshaking

The module uses the following rules when hardware handshaking is enabled. The module:

- does not transmit until CTS becomes active
- examines DSR following the receipt of a character. If the DSR is active, the character is placed in the input queue. If DSR is inactive, the character is assumed to be noise and is discarded.

DTE and DCE Overview

IMPORTANT You need to know whether the device connecting to the module has a DTE or DCE interface. Figure 3.8 through Figure 3.12 are provided to help you make the appropriate connection.

DTE - Data Terminal Equipment

The module's serial ports are configured as 9-pin Data Terminal Equipment (DTE), as are most terminals or computer ports.

Table 3.2 DTE Configurations

DTE 9 Pinout		Signal from DTE Perspective	DTE 25 Pinout	
Pin #	Signal Description		Pin #	Signal Description
1	NC-No Connection (for BASIC module only)	Input	8	CD-Carrier Detect
2	RXD-Received Data	Input	3	
3	TXD-Transmitted Data	Output	2	
4	DTR-Data Terminal Ready	Output	20	
5	Com-Signal Common	Shared	7	
6	DSR-Data Set Ready	Input	6	
7	RTS-Request to Send	Output	4	
8	CTS-Clear to Send	Input	5	
9	NC-No Connection (for BASIC module only)	Input	22	RI-Ring Indicator

DCE - Data Communication Equipment

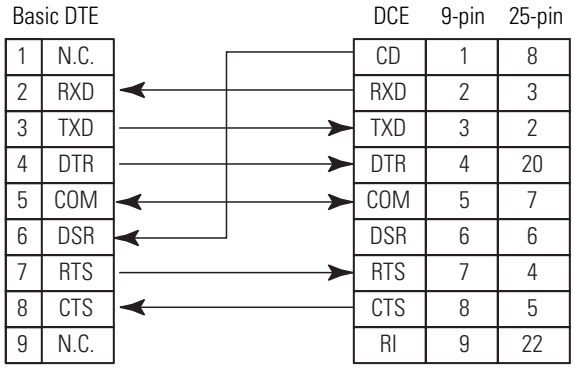
Devices such as modems are Data Communication Equipment (DCE). The pinouts on these terminals are defined for ease of interfacing with DTE equipment.

Table 3.3 DCE Configurations

DCE 9 pinout		Signal from DCE Perspective	DCE 25 pinout
Pin #	Signal Description		Pin #
1	CD-Carrier Detect	Output	8
2	RXD-Received Data	Output	3
3	TXD-Transmitted Data	Input	2
4	DTR-Data Terminal Ready	Input	20
5	Com-Signal Common	Shared	7
6	DSR-Data Set Ready	Output	6
7	RTS-Request to Send	Input	4
8	CTS-Clear to Send	Output	5
9	RI-Ring Indicator	Output	22

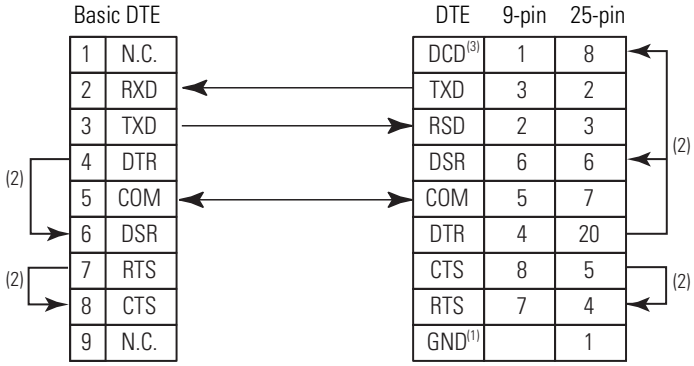
IMPORTANT All signal directions are listed in the previous two tables are valid. For example, TXD, Transmitted Data, is a DTE output but is also a DCE input. The signal description is the same for both the DTE and DCE but the direction of the signal (perspective) has changed based on whether you have a DTE or DCE device.

Figure 3.8 RS-232/423 Wiring Diagram - Module to a Modem (Hardware Handshaking Enabled)



IMPORTANT For DCE devices other than modems, connect the DSR of the module with the DSR of the device. The CD signal of the device (other than a modem) is not used

Figure 3.9 RS-232/423 Wiring Diagram - Module to DTE Device (Hardware Handshaking Disabled)

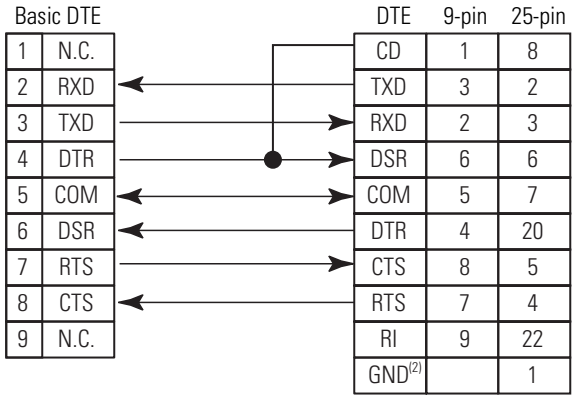


(1) Connect to the shield of the cable.

(2) Jumpers are only needed if you cannot disable the hardware handshaking on the port.

(3) This is a N.C. for the 1747-KE, 1746-BAS or 1746-BAS-T.

Figure 3.10 RS-232/423 Wiring Diagram - Module to Printer (Hardware Handshaking Enabled, Standard Printer Adapter Cable)(1)



(1) The 1747-CP3 Cable works in this application.

(2) Connect to the shield of the cable.

Figure 3.11 RS-422 Wiring Diagram

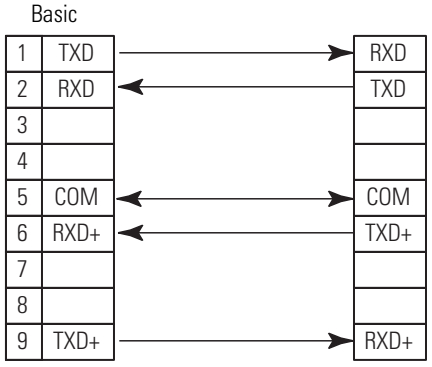
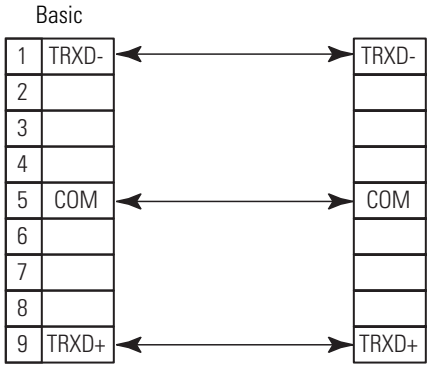


Figure 3.12 RS-485 Wiring Diagram



Wiring to Port DH485

Port DH485 can communicate to user devices through the DH485 communication mode. Use a 1747-C10 Cable or 1747-C13 Cable to connect the module to a link coupler interfaced with the DH485 network.

Programming Overview

This chapter provides an overview of the information needed to program your module. After reading this chapter, you should be familiar with:

- Module memory organization
- SLC memory allocation for your module
- BASIC programming instructions
- entering, running, and editing a BASIC program from an ASCII terminal
- interfacing the module with other devices

Understanding Module Memory Organization

All data transferred to the module from the SLC 500 CPU must be routed through the module input buffer. Table 4.1 lists the addresses of the module input buffer.

Table 4.1 Input Buffer Addresses

Address	Definition
0 through 39	Data transferred from the DH485 common interface file.
40 through 99	Reserved
100 through 163	Data transferred from the CPU M0 file.
164 through 199	Reserved
200 through 207	Data transferred from the CPU output image table.

All data transferred from the module to the SLC 500 CPU must be routed through the module output buffer. Table 4.2 lists the addresses of the module output buffer.

Table 4.2 Output Buffer Addresses

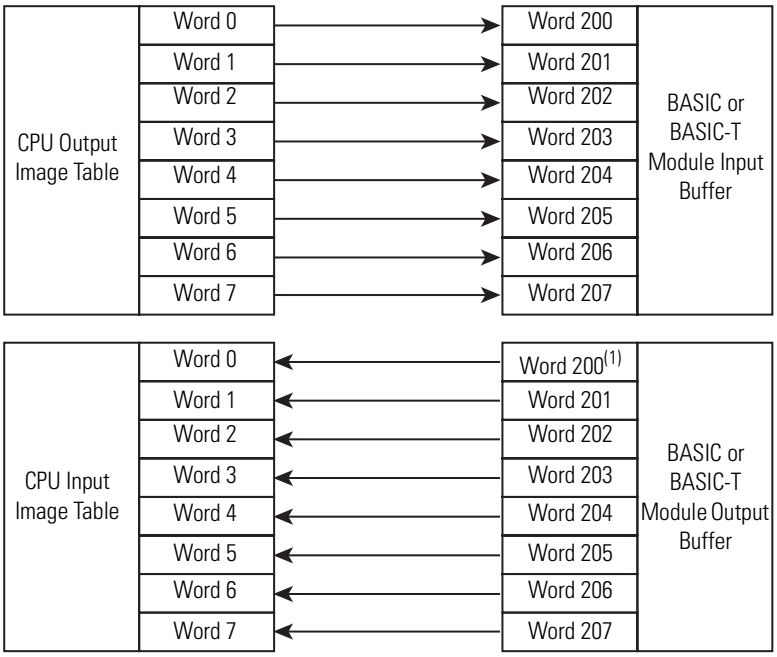
Address	Definition
0 through 39	Data transferred to the DH485 common interface file.
40 through 99	Reserved
100 through 163	Data transferred to the CPU M1 file.
164 through 199	Reserved
200 through 207	Data transferred to the CPU input image table.

Allocating SLC memory for the Module

Your SLC 500 fixed or modular controller communicates to the module through the SLC backplane interface. The backplane interface transfers data from the CPU input and output image tables to the module input and output buffers as shown in Figure 4.1.

For more information regarding the transfer of data between the SLC and the module, refer to page 4-15 of this manual and to the *BASIC Language Reference Manual* (publication number 1747-RM001A-US-P).

Figure 4.1 SLC 500 to Module Data Transfer

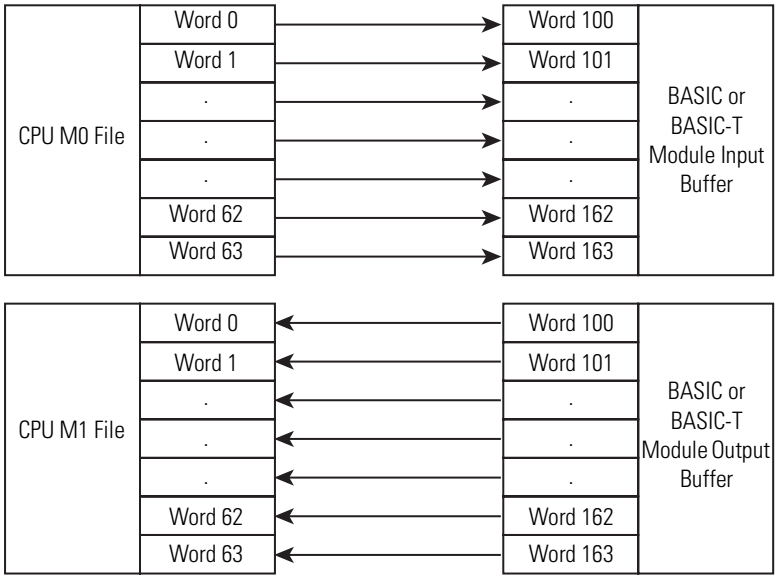


(1) Word 200 is predefined. You cannot write to word 200 of the BASIC output buffer.

In addition to transferring image table data, the SLC 5/02 and above modular controller may transfer an additional 64 words of input or output data to the module. Figure 4.2 shows the transfer of M0 and M1 files between the SLC 5/02 and higher processors and the module.

For more information on M0 and M1 files, refer to the *SLC 500™ and MicroLogix™ 1000 Instruction Set Reference Manual*, publication number 1747-6.15.

Figure 4.2 SLC 5/02 to Module M0 and M1 File Transfer



Module ID Codes

Table 4.3 lists the ID codes needed to configure the memory of your SLC 500 fixed or modular controller.

Table 4.3 Module ID Codes

Controller	Module ID Code	
	1746-BAS	1746-BAS-T
SLC 500, 5/01	3506	3523 (Class 1)
SLC 5/02 and higher	13106	13123 (Class 4)

BASIC Programming Instructions

BASIC programs are composed of BASIC programming instructions grouped together. These instructions are a combination of BASIC commands, statements, operators, and system subroutines (CALLs).

IMPORTANT The module operates in two modes: the Command mode (Direct mode) and the Run mode (Interpreter mode). You can only enter commands when the processor is in the Command mode.

BASIC Commands

BASIC commands are programming instructions that are executed during the Command mode except for CONTROL-C. CONTROL-C takes you from Run mode to Command mode. Typically these commands are used to perform some type of program maintenance. Table 4.4 lists the BASIC programming commands.

Table 4.4 BASIC Commands

Command	Function	Examples
CONT	CONTinue program execution after a STOP statement or CONTROL-C command.	CONT
CONTROL-C	Stop current program execution in Run mode and return module to Command mode.	[CTRL-C]
CONTROL S	Interrupt a LIST command.	[CTRL-S]
CONTROL Q	Restart a LIST command after a CONTROL S command.	[CTRL-Q]
DISABLING CONTROL-C	Disable the CONTROL-C break function. CALL 18 disables the CONTROL-C break function. CALL 19 re-enables the CONTROL-C break function.	CALL 18 (disable) CALL 19 (re-enable)
ERASE	Erase the program stored in ROM.	ERASE
LIST	LIST current program or indicated lines of program to the console device.	LIST, LIST 10-50
LIST#	LIST current program or indicated lines of program to the device connected to port PRT1.	LIST#, LIST#50
LIST@	LIST current program or indicated lines of program to the device connected to port PRT2.	LIST@, LIST@50
MODE	Set up port parameters.	MODE(DH485,19200)
NEW	Erase the program stored in RAM.	NEW
NULL	Count the number of null characters the module outputs after a carriage return.	NULL, NULL4
PROG	Program the EEPROM module with the current program.	PROG
PROG1	Program the EEPROM module with port information for all three ports and store MTOP information.	PROG1
PROG2	Execute the first program stored in EEPROM when the module is turned on.	PROG2
RAM	Select the current program from RAM.	RAM
ROM	Select the current program from EEPROM.	ROM, ROM3
RUN	Execute the currently selected program.	RUN
XFER	Transfer a program from EEPROM to RAM, then select RAM mode.	XFER

Refer to the *BASIC Language Reference Manual* (publication number 1747-RM001A-US-P) for additional information on these commands.

BASIC Statements

BASIC statements are programming instructions that are executed during Run mode. Typically these statements are used to control program execution. Table 4.5 lists the BASIC programming statements.

Table 4.5 BASIC Statements

Statement	Function	Examples
CLEAR	CLEAR variables, interrupts, and strings.	CLEAR
CLEAR (S&I)	CLEAR stacks and interrupts.	CLEARS, CLEARI
CLOCK (1&0)	Enable and disable free running clock.	CLOCK1, CLOCK0
DATA	Read information with the DATA statement.	DATA 100
DIM	Allocate memory for arrayed variables.	DIM A(20)
DO-WHILE	Set up loop for WHILE.	DO - WHILE
DO-UNTIL	Set up loop for UNTIL.	DO - UNTIL
END	Terminate program execution.	END
FOR-TO-STEP	Set up FOR-NEXT loop.	FOR A = 1 TO 5
GOSUB	Execute subroutine.	GOSUB 1000
GOTO	GOTO program line number.	GOTO 500
IF-THEN-ELSE	Test for a condition.	IF A>B THEN A=0
INPUT	INPUT a string or variable.	INPUT A
LD@	Load top of stack from user specified location.	LD@ 1000H, LD@ A
LET	Assign a variable or string a value. (LET is optional.)	LET A=1
NEXT	Test FOR-NEXT loop condition.	NEXT A
ONERR	Perform conditional arithmetic ERROR handling.	ONERROR 10
ON-GOTO	Perform conditional GOTO.	ON A GOTO 5, 20
ON-GOSUB	Perform conditional GOSUB.	ON A GOSUB 6, 2
ONTIME	Generate an interrupt when TIME is equal to or greater than the expression following ONTIME statement.	ONTIME10, 1000
PH0.	PRINT HEX mode with zero suppression.	PH0. A
PH1.	PRINT HEX mode without zero suppression.	PH1. A
POP	POP argument stack to variables.	POP A, B, C
PRINT	PRINT variables, strings or literals.	PRINT A
PUSH	PUSH expressions on argument stack.	PUSH 10, A
READ	READ data in a DATA statement.	READ A
REM	Insert a remark statement in a program.	REM DONE
RESTORE	RESTORE READ pointer.	RESTORE
RETI	RETURN from interrupt.	RETI
RETURN	RETURN from subroutine.	RETURN
ST@	Store top of stack at user specified location.	ST@ 1000H, ST@ A
STOP	Break program execution.	STOP
STRING	Allocate memory for STRINGs.	STRING 50, 10

Refer to the *BASIC Language Reference Manual* (publication number1747-RM001A-US-P) for additional information on these statements.

BASIC Operators

BASIC operators are programming instructions that are executed during Run mode. Typically these operators perform a predefined operation on either variables or constants. Operators require either one or two operands. Table 4.6 lists the BASIC programming operators.

Table 4.6 BASIC Operators

Operator	Function	Examples
ABS ( )	Return the absolute value of expression.	ABS (-3)
( ) + ( )	Add expressions together.	1+1
ASC ( )	Return integer value of ASCII character.	ASC (3)
ATN ( )	Return arraignment of argument.	ATN (1)
CHR ( )	Convert numeric expression to ASCII value.	CHR (65)
COS ( )	Return the cosine of argument.	COS (0)
( ) / ( )	Divide first expression by second expression.	10/2
EOF	Test for empty input buffer.	IF (NOT(EOF))
EXP ( )	Raise number to power of argument.	EXP (10)
( ) ** ( )	Raise first expression by the power of the second expression.	2**4
FREE	List available bytes in RAM.	FREE=
GET	Read console.	P. GET
INT ( )	Return integer portion of expression.	INT (3.2)
IP	Read/assign IP register.	IP=0
LEN	List amount of bytes in current program.	LEN
LOG ( )	Return the natural log of the argument.	LOG (10)
( ) .AND. ( )	Combine the first expression with the second expression using .AND..	10.AND.5
( ) .OR. ( )	Combine the first expression with the second expression using .OR..	2.OR.1
( ) .XOR. ( )	Combine the first expression with the second expression using .XOR..	3.XOR.2
MTOP	Return last valid memory address.	PRINT MTOP
( ) * ( )	Multiply expressions together.	4*4
p	Store constant.	3.1415926
RND	Return a random number.	RND
SGN ( )	Return the sign of argument.	SGN (-5)
SIN ( )	Return the sine of argument.	SIN (3.14)
SQR ( )	Return the square root of the argument.	SQR (100)
( ) - ( )	Subtract one expression from another.	8-4
TAN ( )	Return the tangent of argument.	TAN (.707)
TCON	Read/assign TCON register.	TCON=10H
TIME	Read/assign the free running clock.	P. TIME
XBY ( )	Read/assign external data memory.	P. XBY (10)
( ) = ( )	Allow the first expression to equal the second expression.	10=10

Table 4.6 BASIC Operators

Operator	Function	Examples
( ) < ( )	Allow the first expression to be less than the second expression.	9<10
( ) <= ( )	Allow the first expression to be less than or equal to the second expression.	X<=10
( ) > ( )	Allow the first expression to be greater than the second expression.	10>9
( ) >= ( )	Allow the first expression to be greater than or equal to the second expression.	X>=10
( ) <> ( )	Allows the first expression to be unequal to the second expression.	10<>9

Refer to the *BASIC Language Reference Manual* (publication number 1747-RM001A-US-P) for additional information on these statements.

Creating and Editing a BASIC Program

Module execution is controlled through a BASIC program residing in RAM or ROM. You have the option of creating and editing this program:

- either on a personal computer using the BASIC development software and then downloading it to the module. BASIC development software uses an MS-DOS compatible personal computer to facilitate editing, compiling (translating), uploading, and downloading BASIC programs. Refer to the *BASIC Development Software Programming Manual* (publication number 1747-PM001A-US-P) for additional information on the BASIC development software.
- or entering the program one line at a time directly to the module using an ASCII terminal. ASCII terminal programming must be done one line at a time.

Entering a BASIC Program Using an ASCII Terminal

BASIC line numbers indicate the order in which the program lines are stored in memory. They are also used as references when branching and editing. Typically you start numbering BASIC programs with line number 10 and increment by 10. This allows you to add additional lines later as you work on your program.

Since the computer runs the statements in numerical order, additional lines need not appear in consecutive order on the screen. For example, if you enter line 35 after line 40, the computer still runs line 35 after line 30 and before line 40. This technique saves you from re-entering an entire program if you forget to include a line.

IMPORTANT Reuse of an existing line number causes all of the information referenced by the original line number to be lost. Be careful when entering numbers in the Command mode; you may accidentally erase some program lines.

After the line number, there may be a combination of BASIC commands, statements, operators, or CALLs. See Table 4.4, Table 4.5, and Table 4.6 for a list of BASIC commands, statements, and operators. Depending on the logic of your program, there may be more than one statement on a line. If so, each statement must be separated by a colon (:).

To enter a BASIC program using an ASCII terminal follow these steps:

1. Select the program port using JW4.
2. Connect the ASCII terminal to the selected program port on the module.
3. Verify that the console device is configured to communicate with the module (protocol and communication settings).
4. Apply power to your system.

If there is no program in RAM, this appears on the ASCII terminal.

```
SLC 500 BASIC Module - Catalog Number 1746-BAS
Firmware release: 1.00
Allen-Bradley Company, Copyright 1991
All rights reserved

>
```

If there is a program in RAM and the module has been programmed to execute from RAM, this program starts running. If you type `[Ctrl-C]` this screen appears:

```
.
.
.
STOP - IN LINE XXX
READY
>
```

IMPORTANT The system prompt `[>]` indicates that the module is in Command mode, and the module "ACT" indicator light should be blinking.

5. Enter a line of the BASIC program at the system prompt [**>**].

```
READY
>10 REM FIRST PROGRAM
>20 PRINT "HELLO WORLD"
```

A BASIC program line always begins with a line number and must contain at least one character, but no more than 68 characters.

6. Press **[RETURN]** to end the program line.

Running a BASIC Program

After entering your BASIC program, you are ready to run it. To run a BASIC program, type **RUN** at the system prompt [**>**].

```
READY
>RUN

HELLO WORLD

READY
>
```

Stopping a BASIC Program

To stop a program that is running, press **[Ctrl-C]**.

IMPORTANT If **[Ctrl-C]** is disabled, you cannot stop program execution through a BASIC command. You must have jumper JW4 set in the default position and cycle power to stop program execution. (See page 3-4.)

Editing a BASIC Program Line Through an ASCII Terminal

When the module is in Command mode, you can edit the BASIC program that resides in RAM. Editing a BASIC program is done on a line-by-line basis. To edit an existing line in the BASIC program, type **EDIT** and the line number of the line to edit as shown on the following screen:

```
READY
>EDIT 10
```

The BASIC program line specified by the Edit command is displayed on the ASCII terminal. You can perform any of the following edit operations:

- cursor right and left
- replace a character
- insert a character
- delete a character
- retype a line

Table 4.7 lists the BASIC edit operations, their functions, and keystrokes required to perform the edit operation.

Table 4.7 BASIC Edit Operations

Operation	Use To	Key Strokes
Move	Provide right/left cursor control.	`[Space bar]` – moves the cursor one space to the right. `[Backspace]` – moves the cursor one space to the left.
Replace	Replace the character at the current cursor position.	Press the key that corresponds to the character that will replace the character at the current cursor position.
Insert	Insert text at the current cursor position. Important: When you use the Insert command, all text to the right of the cursor disappears until you type the second `[Ctrl-A]`. Total line length is 79 characters.	`[Ctrl-A]` Important: You must type a second `[Ctrl-A]` to terminate the Insert operation.
Delete	Delete the character at the cursor position.	`[Ctrl-D]`
Exit	Exit the editor with or without saving the changes.	`[Ctrl-Q]` – exits the editor and replaces the old line with the edited line. `[Ctrl-C]` – exits the editor without saving any changes made to the line.
Retype	Copy the current line of text and insert it at the line following the current line. The cursor is moved to the first character on the new line.	`[RETURN]`

Deleting a BASIC Program Line

When the module is in Command mode, you can delete an existing line of the BASIC program. To delete an existing line of the BASIC program, type the line number of the line to delete; then press `[RETURN]` as shown on the following screen:

```
READY
>10
```

Renumbering a BASIC Program

When the module is in Command mode, you can renumber the BASIC program that resides in RAM. To renumber a BASIC program, you must enter a REN command at the system prompt [>]. Table 4.8 lists the commands, functions, and keystrokes needed to renumber your BASIC program.

IMPORTANT

- The REN command updates the destination of GOSUB, GOTO, ON ERR, ONTIME and ON GOTO statements.
- If the target line number does not exist, or if there is insufficient memory to complete the task, no lines are changed and the message `RENUMBER ERROR` appears on the console screen.
- Because the REN command uses the same RAM for renumbering as it does for variable and program storage, available RAM may be insufficient in large programs. You should renumber your program periodically during development instead of waiting until it is completed.

Command	Renumbers the Program Starting at	Key Strokes
Renumber	the beginning of the program. The new line numbers begin at 10 and increment by 10.	`[REN]`
	the beginning of the program. The new line numbers begin at 10 and increment by NUM.	`[REN[NUM]]`
	the beginning of the program. The new line numbers begin with NUM1 and increment by NUM2.	`[REN[NUM1],[NUM2]]`
	NUM2. The new line numbers begin with NUM1 and increment by NUM3.	`[REN[NUM1],[NUM2],[NUM3]]`

Transferring Data

You can transfer data, through the use of various commands, between the SLC processor and:

- port PRT2
- port PRT1
- port DH485
- the 1746-BAS or 1746-BAS-T module

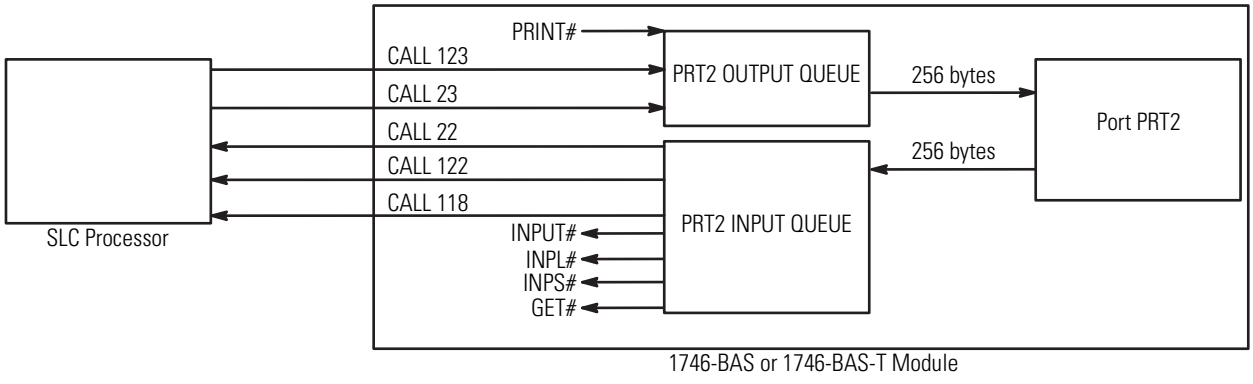
Transferring Data Between the SLC Processor and Port PRT2

Use port PRT2 to interface the module with external devices sending ASCII code (bar code decoders and printers) or DF1 packets (PLCs). The commands in the following table are used to transfer data either to or from port PRT2.

Table 4.8 Data Transfer Commands

Command	Purpose
PRINT#	Prints a string or variable to PRT2.
INPUT#	Inputs a string or variable from PRT2.
INPL#	Inputs a string or variable from PRT2.
INPS#	Inputs a string or variable from PRT2.
GET#	Reads a console input device connected to PRT2.
CALL 22	Transfers data from PRT1 or PRT2 to the SLC I/O or M files.
CALL 23	Transfers data from the SLC I/O or M files to PRT1 or PRT2.
CALL 118	Allows unsolicited writes from a remote SLC or PLC node.
CALL 122	Reads a PLC data file and transfers it to the SLC I/O or M files.
CALL 123	Transfers data from the SLC I/O or M files to a remote PLC.

Figure 4.3 Data Flow Between the SLC Processor and Port PRT2 of the Module



In addition, the commands in the following table provide status of and control over the data transfer between the SLC processor and port PRT2 of the module.

Table 4.9 Status and Control Commands

Command	Purpose
MODE	Sets the port parameters of PRT1, PRT2, and DH485.
CALL 16	Enables interrupt capability when a DF1 packet is received.
CALL 17	Disables the DF1 packet interrupt capability.
CALL 30	Sets the port parameters for PRT2.
CALL 31	Displays the current PRT2 port configuration on the program port terminal screen.
CALL 35	Retrieves the current character in the 256 character input buffer of port PRT2.
CALL 36	Retrieves the number of characters in the input or output buffer of port PRT2.
CALL 37	Clears the peripheral port input and/or output buffers.
CALL 97	Enables the DTR signal for port PRT2.
CALL 98	Disables the DTR signal for port PRT2.
CALL 108	Enables DF1 driver communications. You must use this CALL in conjunction with CALLs 16, 17, 118, 122, and 123.
CALL 110	Prints the complete output buffer with addresses, front pointer, and number of characters in the buffer to the console device.
CALL 111	Prints the complete input buffer with addresses, front pointer, and number of characters in the buffer to the console device.
CALL 113	Disables DF1 driver communications.
CALL 114	Initiates DF1 packet transmission.
CALL 115	Checks DF1 packet transmission status.
CALL 117	Gets DF1 packet length.
CALL 119	Resets port parameters back to their default settings.

For more information regarding the use of these commands, refer to the *BASIC Language Reference Manual* (publication number 1747-RM001A-US-P).

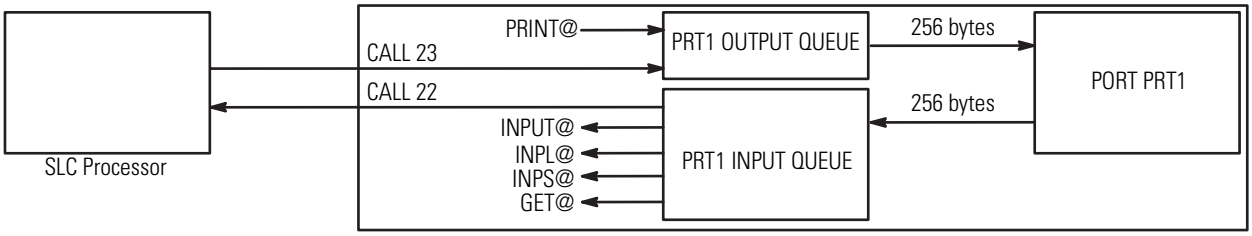
Transferring Data Between the SLC Processor and Port PRT1

Use port PRT1 to interface the module with external devices sending ASCII code. The commands in the following table are used to transfer data either to or from port PRT1.

Table 4.10 Data Transfer Commands

Command	Purpose
PRINT@	Prints a string or variable to PRT1.
INPUT@	Inputs a string or variable from PRT1.
INPL@	Inputs a string or variable from PRT1.
INPS@	Inputs a string or variable from PRT1.
GET@	Reads a console input device connected to PRT1.
CALL 22	Transfers data from PRT1 or PRT2 to the SLC I/O or M files.
CALL 23	Transfers data from the SLC I/O or M files to PRT1 or PRT2.

Figure 4.4 Data Flow Between the SLC Processor and Port PRT1 of the Module



In addition, the commands in the following table provide status of and control over the data transfer between the SLC processor and port PRT1 of the module.

Table 4.11 Status and Control Commands

Command	Purpose
MODE	Sets the port parameters of PRT1, PRT2, and DH485.
CALL 94	Displays the current port PRT1 configuration on the program port terminal screen.
CALL 95	Retrieves the number of characters in the input or output buffer of port PRT1.
CALL 96	Clears port PRT1 input and output buffers.
CALL 103	Prints the complete output buffer with addresses, front pointer, and number of characters in the buffer to the program port screen.
CALL 104	Prints the complete input buffer with addresses, front pointer, and number of characters in the buffer to the program port screen.
CALL 105	Resets the port parameters of port PRT1 to their default setting.

For more information regarding the use of these commands, refer to the *BASIC Language Reference Manual* (publication number 1747-RM001A-US-P).

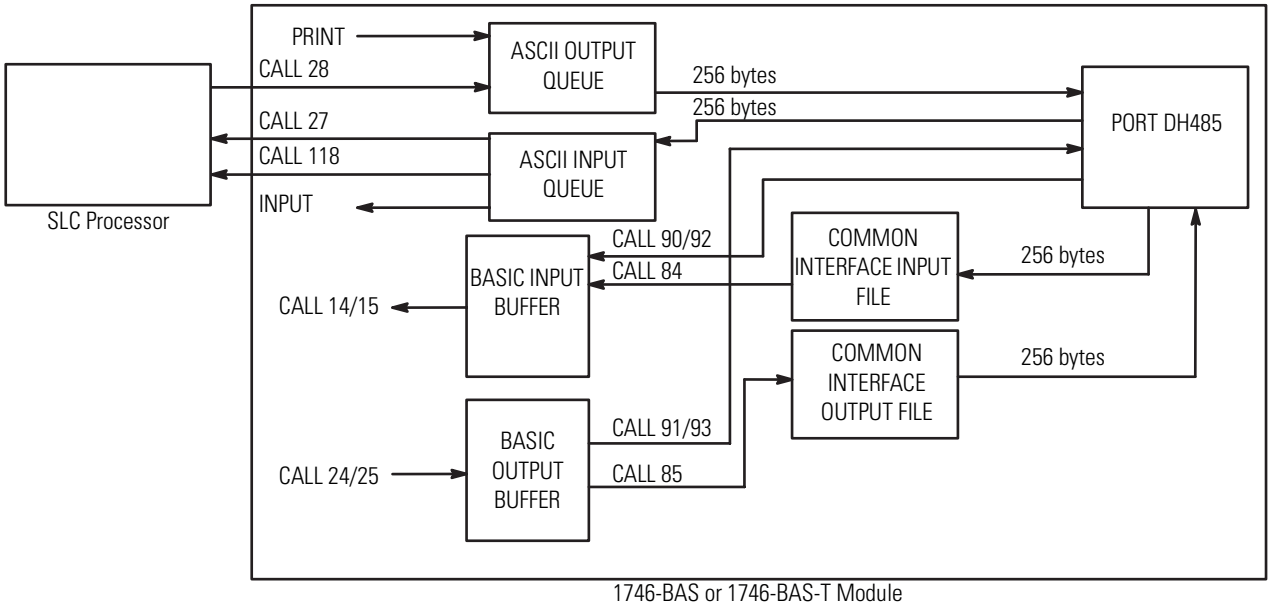
Transferring Data Between the SLC Processor and Port DH485

Use port DH485 to interface the module with the DH485 network (other SLC processors). The commands in the following table are used to transfer data either to or from port DH485.

Table 4.12 Data Transfer Commands

Command	Purpose
CALL 14	Converts 16-bit signed integer located in the BASIC input buffer to BASIC floating-point.
CALL 15	Converts 16-bit unsigned integer located in the BASIC input buffer to BASIC floating-point.
CALL 24	Converts BASIC floating point to a 16-bit signed integer and places the result in the BASIC output buffer.
CALL 25	Converts BASIC floating point to a 16-bit binary number and places the result in BASIC output buffer.
CALL 27	Transfers the data from a remote DH485 data file to the SLC processor.
CALL 28	Transfers the data from the SLC processor to a remote DH485 data file.
CALL 84	Transfers the data from the DH485 common interface file to words 0 through 39 of the module input buffer.
CALL 85	Transfers words 0 through 39 of the module output buffer to the DH485 common interface file.
CALL 90	Transfers the data from a remote DH485 data file to words 0 through 39 of the module input buffer.
CALL 91	Transfers words 0 through 39 of the module output buffer to a remote DH485 data file.
CALL 92	Transfers the data from a remote DH485 interface file to words 0 through 39 of the module input buffer.
CALL 93	Transfers words 0 through 39 of the module output buffer to a remote DH485 interface file.
CALL 118	Allows unsolicited writes from a remote SLC or PLC node.

Figure 4.5 Data Flow Between the SLC processor and Port DH485 of the Module



In addition, the commands in the following table provide status of the data transfer between the SLC processor and port DH485 of the module.

Table 4.13 Status and Control Commands

Command	Purpose
MODE	Sets the port parameters of PRT1, PRT2, and DH485.
CALL 86	Checks the remote write status of the DH485 common interface file.
CALL 87	Checks the remote read status of the DH485 common interface file.

For more information regarding the use of these commands, refer to the *BASIC Language Reference Manual* (publication number 1747-RM001A-US-P).

Transferring Data Between the SLC Processor and the Module

Use the module to interface with the SLC processor. For example, the module performs large mathematical calculations for the processor which the SLC processor uses to execute an operation. The commands in the following table are used to transfer data either to or from the SLC processor.

Table 4.14 Data Transfer Commands

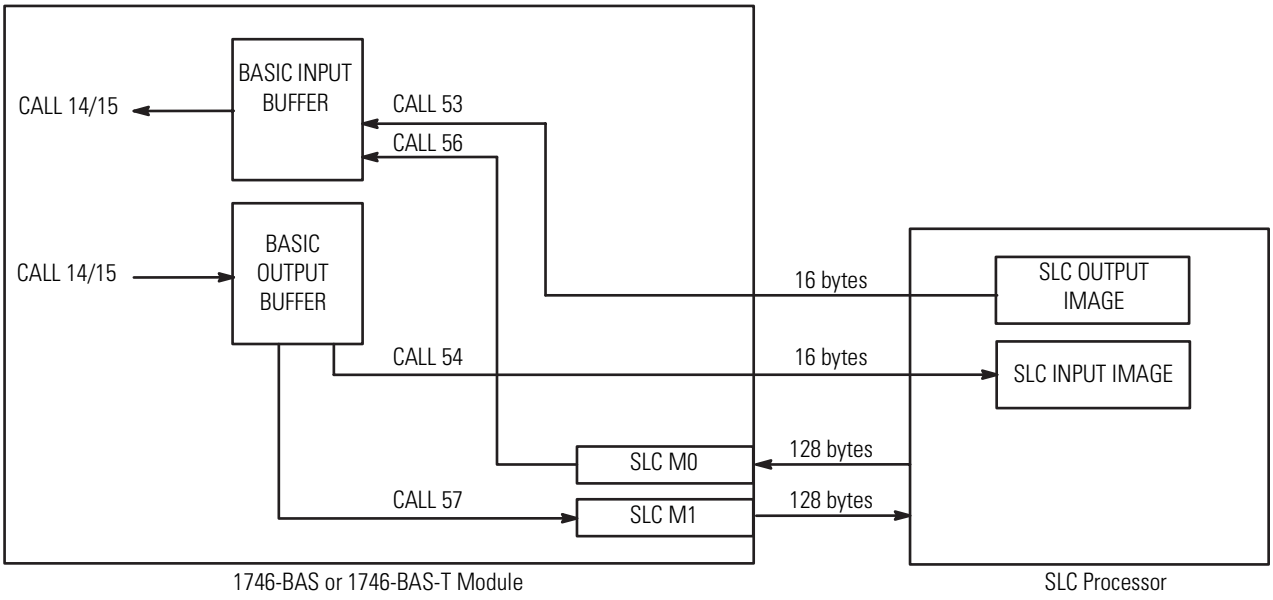
Command	Purpose
CALL 14	Converts 16-bit signed integer located in the BASIC input buffer to BASIC floating-point.
CALL 15	Converts 16-bit unsigned integer located in the BASIC input buffer to BASIC floating-point.
CALL 24	Converts BASIC floating point to a 16-bit signed integer and places the result in the BASIC output buffer.
CALL 25	Converts BASIC floating point to its 16-bit binary representation.
CALL 53	Transfers the eight words in the CPU output image table to words 200 through 207 of the module input buffer.
CALL 54	Transfers words 200 through 207 of the module output buffer to the CPU input image table.
CALL 56	Transfers the words in the CPU M0 file to words 100 through 163 of the module input buffer.
CALL 57	Transfers words 100 through 163 of the module output buffer to the CPU M1 file.

Status Information for the SLC Processor

Input image word 0 for the module slot contains two status bits. One status bit informs the SLC processor of the mode the module is presently in. The other status bit informs the SLC processor of the battery status. These status bits are as follows:

- I:e. 0/13 = 0battery OK
  = 1 battery low
- I:e. 0/15 = 0module in Run mode
  = 1 module in Command mode

Figure 4.6 Data Flow Between the Module and SLC Processor



In addition, the commands in the following table provide status of and control over the data transfer between the SLC processor and module.

Table 4.15 Status and Control Commands

Command	Purpose
CALL 51	Checks if the CPU output image buffer was updated.
CALL 55	Checks if the CPU input image buffer was read by the processor.
CALL 58	Checks if the CPU M0 file was updated.
CALL 59	Checks if the CPU M1 file was read by the processor.
CALL 86	Checks if the DH485 interface file was updated.
CALL 87	Checks if the DH485 interface file was read by an external device.
CALL 120	Clears the module input and output buffers.

For more information regarding the use of these commands, refer to the *BASIC Language Reference Manual* (publication number 1747-RM001A-US-P).

The following table lists module buffer addresses. Refer to page 4-1 for more information regarding module buffer addresses.

Table 4.16 BASIC Input/Output Buffer Address Map

Address	BASIC Input/Output Buffer Address
M1:e.s	100 through 163
M0:e.s	100 through 163
I:e.s	200 through 207
O:e.s	200 through 207
CIF in	0 through 39
CIF out	0 through 39

The SLC processor and module operate independently of each other. The following CALLs allow the SLC processor and module to interrupt each other.

Table 4.17 Interrupt CALLs

CALL	Purpose
CALL 16	Enables interrupt capability when a DF1 packet is received.
CALL 17	Disables the DF1 packet interrupt capability.
CALL 20	Enables SLC processor interrupt capability.
CALL 21	Disables SLC processor interrupt capability.
CALL 26	Generates an interrupt to the SLC processor.

SLC Fault Codes

Fault codes are reported in word 6 of the SLC processor status file. The format of the status word and applicable error codes are shown below:

Figure 4.7 SLC Fault Code Placement

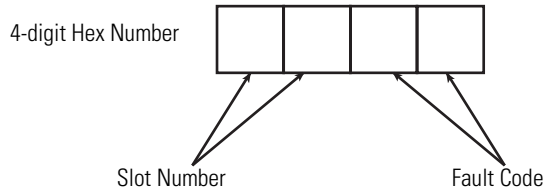


Table 4.18 SLC Fault Codes

SLC Fault Code	Description	Possible Cause	Recommended Action
57H	Module has not responded to a lock shared memory command within the required time limit.	Module hardware problem.	Cycle power to the module to re-initialize parameters and re-run the program.
		Module internal stacks, pointers, etc. (if XBY instructions are used) are corrupted by the user program.	
58H	Module generated a generic fault.	Module hardware problem.	Verify that module is configured correctly (correct I/O and M files). Cycle power to the module to re-initialize parameters and re-run the program.
		Module internal stacks, pointers, etc. (if XBY instructions are used) are corrupted by the user program.	
59H	Module did not complete a command within the required time limit.	Module hardware problem.	Cycle power to the module to re-initialize parameters and re-run the program.
		Module internal stacks, pointers, etc. (if XBY instructions are used) are corrupted by the user program.	

Table 4.18 SLC Fault Codes

SLC Fault Code	Description	Possible Cause	Recommended Action
5AH	Hardware interrupt problem.	Module hardware problem.	Verify that the module slot is enabled. Cycle power to the module to re-initialize parameters and re-run the program.
		Module internal stacks, pointers, etc. (if XBY instructions are used) are corrupted by the user program.	
5BH	G file configuration error.	Module is trying to be configured using G files.	Verify the module is not configured with G files.
5CH	M0/M1 file configuration error.	Wrong M0/M1 file size is chosen.	Verify the module is configured with M0 or M1 files no larger than 64 words.
5DH	Interrupt requested but the processor will not support the interrupt.	SLC 500 fixed controller or 5/01 modular processor does not support module interrupts.	A 5/02 or higher processor must be used for interrupt capability.
75H	Module watchdog timeout.	Module hardware problem.	Cycle power to the module to re-initialize parameters and re-run the program.
		Module internal stacks, pointers, etc. (if XBY instructions are used) are corrupted by the user program.	
90H	Module issued an interrupt while the slot was disabled.	Module is issuing an I/O interrupt. CALL 26 causes this interrupt.	Enable the slot before using interrupts.
SLC Fault Code	Description	Possible Cause	Recommended Action
91H	Module has faulted while slot was disabled.	Module hardware problem.	Cycle power to the module to re-initialize parameters and re-run the program.
		Module internal stacks, pointers, etc. (if XBY instructions are used) are corrupted by the user program.	
92H	Module I/O or ISR configuration is incorrect.	Module slot was configured incorrectly in the SLC ladder logic program.	Verify the slot configuration for the module.
93H	Processor did not recognize the error code from the Module.	Module hardware problem.	Cycle power to the module to re-initialize parameters and re-run the program.
		Module internal stacks, pointers, etc. (if XBY instructions are used) are corrupted by the user program.	

Application errors such as divide by zero error, syntax error, receipt of a CONTROL-C, and execution of STOP or END statements cause the module to return to the Command mode from Run mode. Use CALL 38 (EXPANDED ONERR) to jump to an interrupt routine instead of returning to the Command mode.

Specifications

Module Hardware Specifications

The module hardware specifications are listed in the following tables.

Table A.1 Power Consumption

Operating Voltage	Current Requirement Module Only	Current Requirement Module With Link Coupler
5V dc	.150 A	.150 A
24V dc	.040 A	.125 A

IMPORTANT If a Hand-Held Terminal, Data Table Access Module, or interface converter is connected to the link coupler, the additional backplane power draw of these components (shown in Table A.1) must be added to the .125 Amperes listed in the table above. This only applies when the module is connected to the network via the link coupler and 1747-C10 Cable or 1747-C11 Cable. This does not apply when the 1747-C13 Cable is used.

Table A.2 Power Consumption of Hand-Held Terminal, Data Table Access Module, and Interface Converter

Component	Operating Voltage	Current Requirement
Hand-Held Terminal	24V dc	.105A
Data Table Access Module	24V dc	.104A
Interface Converter	24V dc	.060A

IMPORTANT The BASIC module receives its power from the SLC backplane. The power consumption of the module must be taken into consideration when planning your SLC 500 system. Refer to the documentation supplied with your SLC 500 fixed or modular controller for additional information on power supplies and current requirements.

Table A.3 Environmental Conditions

Condition	Range
Operating temperature	0° C to 60° C (32° F to 140° F)
Storage temperature	-40° C to 85° C (-40° F to 185° F)
Relative humidity	5% to 95% (non-condensing)

Table A.4 Port Isolation

Port	Isolation	Isolation Voltage
PRT1	Backplane to Port	710V dc for 1 minute
PRT2	Backplane to Port	710V dc for 1 minute
PRT1 and PRT2	PRT1 to PRT2	710V dc for 1 minute

IMPORTANT Port DH485 is not isolated.

Table A.5 Clock/Calendar Accuracy

Specification	Range
Accuracy	± 1 minute/month @ 25°C
	+ 0, - 6 minute/month @ 60°C

Table A.6 Maximum Communication Distances

Communication	Maximum Distance Allowed m (ft)			
Rate (bps)	RS-232	RS-423	RS-422	RS-485
300	15 (50)	1230 (4000)	1230 (4000)	1230 (4000)
600	15 (50)	920 (3000)	1230 (4000)	1230 (4000)
1200	15 (50)	770 (2500)	1230 (4000)	1230 (4000)
4800	15 (50)	245 (800)	1230 (4000)	1230 (4000)
9600	15 (50)	120 (400)	1230 (4000)	1230 (4000)
19200	15 (50)	60 (200)	1230 (4000)	1230 (4000)

IMPORTANT Use the RS-423 jumper settings when communicating in RS-232 mode.

1747-PBASE BASIC Development Software Specifications

The BASIC Development Software must be loaded into a personal computer to operate. This personal computer must conform to the following specifications:

- IBM PC/AT compatible computer with display and keyboard
- DOS version 3.1 to 6.22
- 640K bytes of RAM memory
- 1 floppy disk drive (3 1/2 in. or 5 1/4 in.)
- hard disk with 2M bytes free disk space
- 1 RS-232 compatible serial port

Refer to the *BASIC Development Software Programming Manual* (publication number 1747-PM001A-US-P) for additional information.

Related Products

Table 1.H lists the products related to the module.

Table A.7 BASIC Module Related Products

Product	Catalog Number
8K byte EEPROM Memory Module for 1746-BAS	1747-M1
32K byte EEPROM Memory Module for 1746-BAS	1747-M2
8K byte UVPROM Memory Module for 1746-BAS	1747-M3
32K byte UVPROM Memory Module for 1746-BAS	1747-M4
8K byte EEPROM Memory Module for 1746-BAS-T	1771-DBMEM1
32K byte EEPROM Memory Module for 1746-BAS-T	1771-DBMEM2
BASIC Development Software	1747-PBASE
Communication Cable (72" length, interchangeable with C-11 cable)	1747-C10
Communication Cable (12" length, interchangeable with C-10 cable)	1747-C11
Communication Cable (36" length, different from C-10 and C-11 cables)	1747-C13
DH485 Interface Card	1784-KR
Interface/Converter (RS-232 to RS-485)	1747-PIC
Link Coupler	1747-AIC
SLC 500	1747-L20, -L30, -L40
SLC 5/01	1747-L511, -L514
SLC 5/02	1747-L524
SLC 5/03	1747-L531, -L532
SLC 5/04	1747-L541, -L542, -L543
SLC 5/05	1747-L551, -L552, -L553

This appendix contains important information you should be concerned with when configuring the module. The information is general in nature and supplements specific information contained in earlier chapters of this manual.

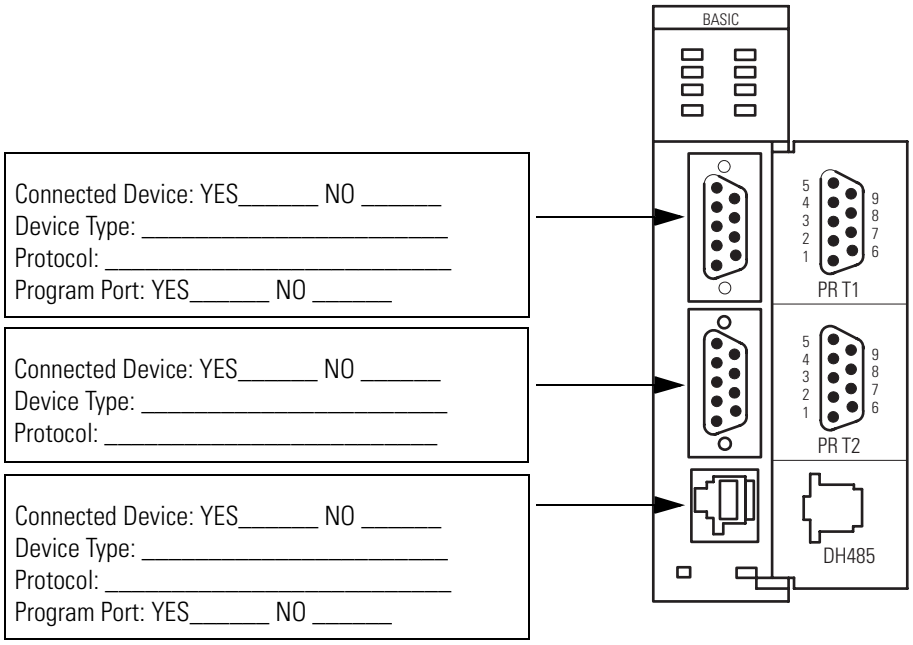
Topics include worksheets for configuring:

- the BASIC or BASIC-T module
- jumpers JW1-JW4

Module Configuration

Specify the connection information by filling in the boxes in the figure below.

Figure 2.1 Module Configuration



IMPORTANT When DF1 protocol is selected for port PRT2, the DH485 port is not available for programming or run time operation.

What to Do Next: Give a copy of this worksheet to the hardware installer. Store this worksheet with your application program for future reference.

Port PRT1 Configuration (Jumper JW1)

See Figure 3.1 on page 3-1 for the locations of the four jumpers.

See Figure 3.2 on page 3-2 for jumper JW1 pin assignments and settings.

Specify the configuration of port PRT1 on your module by filling in the table below.

Table B.1 Port PRT1 Configuration

Port	Configuration	Selection	Corresponding Jumper Position on JW1
PRT1	RS-232/423		Across pins 1 and 2, 3 and 4
	RS-422		Across pins 5 and 6, 7 and 8
	RS-485		Across pins 7 and 8, 9 and 10

What to Do Next: Give a copy of this worksheet to the hardware installer. Store this worksheet with your application program for future reference.

Port PRT2 Configuration (Jumper JW2)

See Figure 3.1 on page 3-1 for the locations of the four jumpers.

See Figure 3.3 on page 3-3 for jumper JW2 pin assignments and settings.

Specify the configuration of port PRT2 on your module by filling in the table below.

Table B.2 Port PRT2 Configuration

Port	Configuration	Selection	Corresponding Jumper Position on JW2
PRT2	RS-232/423		Across pins 1 and 2, 3 and 4
	RS-422		Across pins 5 and 6, 7 and 8
	RS-485		Across pins 7 and 8, 9 and 10

What to Do Next: Give a copy of this worksheet to the hardware installer. Store this worksheet with your application program for future reference.

Optional Memory Module Selection (Jumper JW3)

See Figure 3.1 on page 3-1 for the locations of the four jumpers.

See Figure 3.4 on page 3-4 for jumper JW3 pin assignments and settings.

Specify the optional memory module selection for the system by filling in the table below.

Table B.3 Optional Memory Module Selection

Memory Module Option	Optional Memory Module Selection	Corresponding Jumper Position on JW3
1747-M1 8K byte EEPROM (1746-BAS only)		Across pins 1 and 3, 2 and 4
1747-M2 32K byte EEPROM (1746-BAS only)		Across pins 1 and 3, 2 and 4
1747-M3 8K byte UVPROM (1746-BAS only)		Across pins 1 and 3, 2 and 4
1747-M4 32K byte UVPROM (1746-BAS only)		Across pins 3 and 5, 4 and 6
1771-DBMEM1 8K byte EEPROM (1746-BAS-T only)		Across pins 1 and 3, 2 and 4
1771-DBMEM 32K byte EEPROM (1746-BAS-T only)		Across pins 1 and 3, 2 and 4

What to Do Next: Give a copy of this worksheet to the hardware installer. Store this worksheet with your application program for future reference.

Program Port and Protocol Selection (Jumper JW4)

See Figure 3.1 on page 3-1 for the locations of the four jumpers.

See Figure 3.5 on page 3-5 for jumper JW4 pin assignments and settings.

Specify the Program Port and its protocol by filling in the table below.

Table B.4 Program Port and Protocol Selection

Selection	Port	Program Port?	Protocol	Corresponding Jumper Position on JW4
	PRT1	YES	Default Communication Settings	Across pins 3 and 4, 5 and 6
	PRT2	NO	ASCII Interface	
	DH485	NO	Run Time DH485	
	PRT1	NO	ASCII Interface	Across pins 1 and 3, 2 and 4
	PRT2	NO	ASCII Interface	
	DH485	YES	DH485 Program Port (non-isolated)	
	PRT1	YES	Programmed Communication Settings	Across pins 3 and 5, 4 and 6
	PRT2	NO	ASCII Interface	
	DH485	NO	Run Time DH485	
	PRT1	YES	Programmed Communication Settings	Across pins 1 and 2, 3 and 4
	PRT2	NO	DF1	
	DH485	NO	Disabled	

What to Do Next: Give a copy of this worksheet to the hardware installer. Store this worksheet with your application program for future reference.

Lithium Battery Replacement, Handling, and Disposal

This appendix contains important information you should know when using lithium batteries.

Topics include:

- battery replacement
- battery handling
- battery disposal

Battery Replacement

Your module provides back-up power for RAM through a replaceable lithium battery (catalog number 1747-BA). This battery provides back-up for approximately five years. A BAT LOW indicator on the front of the module alerts you when the battery voltage has fallen below the the replace battery threshold level.

To replace the lithium battery follow these steps:

1. Remove power from the SLC 500 power supply module.

ATTENTION

Do not remove the module from the SLC 500 chassis until all power is removed from the SLC 500 power supply.

2. Remove the module from the chassis by depressing the retainer clips at both the top and bottom of the module and slide it out.

IMPORTANT If the top or bottom retainer clips are broken when removing the module they can be easily replaced. Pry the broken clip(s) off from the bottom with a screwdriver, if necessary. Do not twist off. Snap in the replacement clip. Order Catalog Number 1746-R15 (2 per package).

ATTENTION

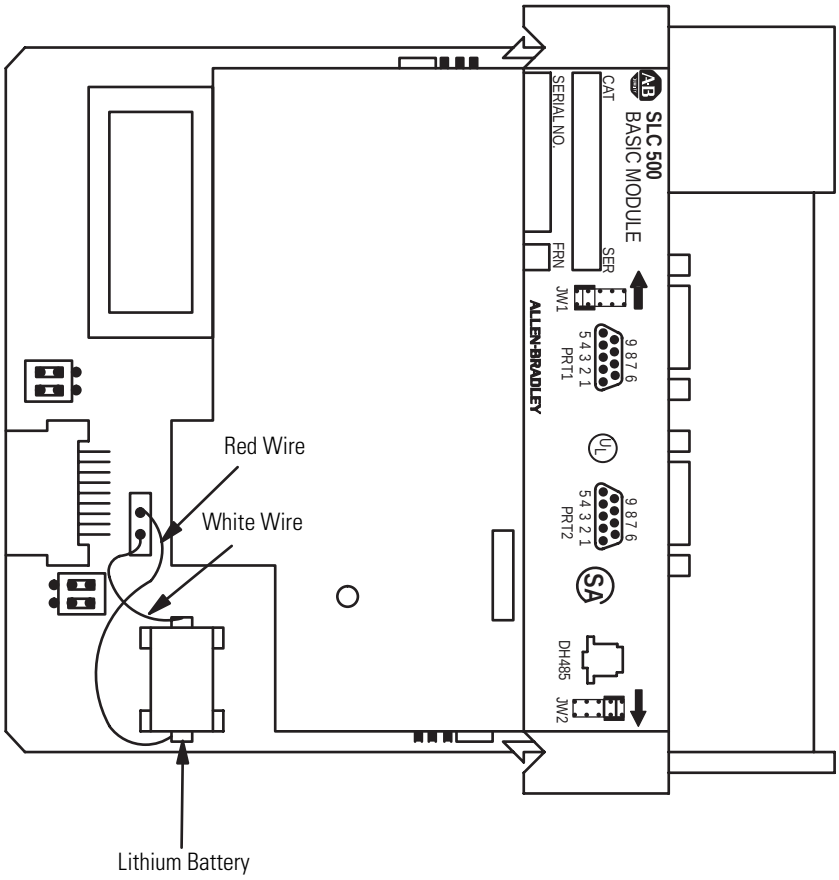
Do not expose the module to surfaces or other areas that may typically hold an electrostatic charge. Electrostatic charges can alter or destroy memory.

3. Unplug the battery connector. See Figure 3.1 on page C-2 for battery connector location.

IMPORTANT The module has a capacitor that provides 30 minutes of battery back-up while the battery is disconnected. Data in RAM is not lost if the battery is replaced within 30 minutes.

4. Remove the battery from the retaining clips.
5. Insert a new battery into the battery retaining clips.
6. Plug the battery connector into the socket with the red lead wire on top and the white lead wire on the bottom. See Figure 3.1 on page C-2 for battery connector orientation.
7. Re-insert the BASIC module into the SLC 500 chassis.
8. Restore power to the SLC 500 power supply module.

Figure 3.1 Lithium Battery



Battery Handling

The procedures listed below must be followed to ensure proper battery operation and reduce personnel hazards:

- Use battery only for intended operation.
- Do not ship or dispose of cells except according to recommended procedures.
- Do not ship on passenger aircraft.

ATTENTION

Do not charge the batteries. An explosion could result or the cells could overheat causing burns.

Do not open, puncture, crush, or otherwise mutilate the batteries. A possibility of an explosion exists and toxic, corrosive, and flammable liquids would be exposed.

Do not incinerate or expose the batteries to high temperatures.

Do not attempt to solder batteries. An explosion could result.

Do not short positive and negative terminals together. Excessive heat can build up and cause severe burns.

Storage

Store lithium batteries in a cool, dry environment, typically +20°C to +25°C (+68°F to +77°F) with 40% to 60% humidity. Store the batteries and a copy of the battery instruction sheet in the original container, away from flammable materials.

Transportation

One or Two Batteries - Each battery contains 0.23 grams of lithium. Therefore, up to two batteries can be shipped together within the United States without restriction. Regulations governing shipment to or within other countries may differ.

Three or More Batteries - Procedures for the transportation of three or more batteries shipped together within the United States are specified by the Department of Transportation (DOT) in the Code of Federal Regulations, CRF49, “Transportation”. An exemption to these regulations, DOT – E7052, covers the transport of certain hazardous materials classified as flammable solids. This exemption authorizes transport of lithium batteries by motor vehicle, rail freight, cargo vessel, and cargo-only aircraft, providing certain conditions are met. Transport by passenger aircraft is not permitted.

A special provision of the DOT – E7052 (11th Rev., October 21, 1982, par. 8-a) provides that:

> Persons that receive cell and batteries covered by this exemption may reship them pursuant to the provisions of 49 CFR 173.22a in any of these packages authorized in this exemption including those in which they were received.

The Code of Federal Regulations, 49 CRF 173.22a, relates to the use of packaging authorized under exemptions. In part, it requires that you must maintain a copy of the exemption at each facility where the packaging is being used in connection with shipment under the exemption.

Shipment of depleted batteries for disposal may be subject to specific regulation of the countries involved or to regulations endorsed by those countries, such as the IATA Restricted Articles Regulations of the International Air Transport Association, Geneva, Switzerland.

Regulations for transportation of lithium batteries are periodically revised.

Battery Disposal

The following procedures must be followed when disposing of lithium batteries.

ATTENTION Do not incinerate or dispose of lithium batteries in general trash collection. Explosion or violent rupture is possible. Batteries should be collected for disposal in a manner to rpevent against short circuiting, compacting, or destruction of case integrity and hermetic seal.

For disposal, batteries must be packaged and shipped in accordance with the transportation regulations, to a proper disposal site. The U.S. Department of Transportation authorizes shipment of "Lithium batteries for disposal" by motor vehicle only in regulation 173.1015 of CRF49 (effective January 5, 1983). For additional information contact:

U.S. Department of Transportation
Research and Special Programs Administration
400 Seventh Street, S.W.
Washington, D.C. 20590

Although the Environmental Protection Agency at this time has no regulations specific to lithium batteries, the material contained may be considered toxic, reactive, or corrosive. The person disposing of the material is responsible for any hazards created in doing so. State and local regulations may exist regarding the disposal of these materials.

Numerics

A

B

C

D

E

H

I

J

L

M

N

O

P

R

S

T

U

W

Reach us now at www.rockwellautomation.com

Wherever you need us, Rockwell Automation brings together leading brands in industrial automation including Allen-Bradley controls, Reliance Electric power transmission products, Dodge mechanical power transmission components, and Rockwell Software. Rockwell Automation's unique, flexible approach to helping customers achieve a competitive advantage is supported by thousands of authorized partners, distributors and system integrators around the world.

Americas Headquarters, 1201 South Second Street, Milwaukee, WI 53204, USA, Tel: (1) 414 382-2000, Fax: (1) 414 382-4444
European Headquarters SA/NV, avenue Herrmann Debroux, 46, 1160 Brussels, Belgium, Tel: (32) 2 663 06 00, Fax: (32) 2 663 06 40
Asia Pacific Headquarters, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication 1746-UM004A-US-P - April 2000
Supersedes Publication 1746-6.1 - November 1994